



## City Research Online

### City, University of London Institutional Repository

---

**Citation:** Calado, J.M.F. (1996). On-line fault diagnosis of industrial processes based on artificial intelligence techniques. (Unpublished Doctoral thesis, City University London)

This is the accepted version of the paper.

This version of the publication may differ from the final published version.

---

**Permanent repository link:** <https://openaccess.city.ac.uk/id/eprint/7773/>

**Link to published version:**

**Copyright:** City Research Online aims to make research outputs of City, University of London available to a wider audience. Copyright and Moral Rights remain with the author(s) and/or copyright holders. URLs from City Research Online may be freely distributed and linked to.

**Reuse:** Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

**ON-LINE FAULT DIAGNOSIS OF  
INDUSTRIAL PROCESSES BASED ON  
ARTIFICIAL INTELLIGENCE TECHNIQUES**

**BY**

**JOÃO MANUEL FERREIRA CALADO**

**A THESIS SUBMITTED FOR THE  
DEGREE OF  
DOCTOR OF PHILOSOPHY**

**CONTROL ENGINEERING RESEARCH CENTRE  
DEPARTMENT OF ELECTRICAL ELECTRONIC AND  
INFORMATION ENGINEERING  
CITY UNIVERSITY  
LONDON**

**MAY 1996**

# Table of Contents

<b>List of Tables</b>	<b>6</b>
<b>List of Figures</b>	<b>7</b>
<b>Acknowledgements</b>	<b>10</b>
<b>Declaration</b>	<b>11</b>
<b>Publications</b>	<b>12</b>
<b>Abstract</b>	<b>14</b>
<b>1. Introduction</b>	<b>15</b>
1.1 Motivation	15
1.2 A Rule Based Controller	17
1.3 Fault Diagnosis Based on Fuzzy Qualitative Modelling	17
1.4 Self-Learning Fault Diagnosis Based on Fuzzy Qualitative Modelling	19
1.5 Process Fault Diagnosis Using a Knowledge Based System Coupled with a Fuzzy Neural Network	20
<b>2. Knowledge Based Systems</b>	<b>23</b>
2.1 Introduction	23
2.2 Production Systems	25
2.3 Knowledge Acquisition	30

2.4 Knowledge Representation	31
2.5 Inference Engine	33
<b>3. Fuzzy Systems</b>	<b>37</b>
3.1 Introduction	37
3.2 Formal Definitions for Fuzzy Sets	39
3.3 Fuzzy Quantity Spaces	44
3.4 The Extension Principle	47
<b>4. Modelling and Rule Based Control of a Mixing Process</b>	<b>51</b>
4.1 Introduction	51
4.2 The Mixing Process	52
4.3 The Mixing Process Dynamic Model	53
4.4 Qualitative Modelling of the Mixing Process	57
4.5 Rule Based Control of the Mixing Process	58
4.6 Performance of the Rule Based Controller	63
4.7 Conclusions	64
<b>5. Modelling and Control of a Continuous Stirred Tank Reactor</b>	<b>66</b>
5.1 Introduction	66
5.2 Continuous Stirred Tank Reactor	67
5.3 The Continuous Stirred Tank Reactor Dynamic Model	68
5.4 Qualitative Modelling of the Continuous Stirred Tank Reactor	73
5.5 Process Variables Control	77
5.6 Controllers Performance	79
5.7 Conclusions	80



<b>6. On-Line Single Fault Diagnosis Based on Fuzzy Qualitative Simulation</b>	<b>82</b>
6.1 Introduction	82
6.2 System Architecture	85
6.3 Fuzzy Qualitative Reasoning	89
6.4 Fault Detection Reasoning	94
6.5 Fault Diagnosis Reasoning	95
6.6 Fault Detection and Diagnosis of the Mixing Process	97
6.7 Conclusions	100
<b>7. On-Line Multiple Faults Diagnosis Through Fuzzy Qualitative Simulation</b>	<b>102</b>
7.1 Introduction	102
7.2 System Architecture	104
7.3 Fault Diagnosis Reasoning	108
7.4 Fault Detection and Diagnosis of the Mixing Process	110
7.5 Faults Detection and Diagnosis of the Continuous Stirred Tank Reactor	114
7.6 Conclusions	123
<b>8. Fuzzy Qualitative Simulation Based Fault Diagnosis with Self-Learning Abilities</b>	<b>125</b>
8.1 Introduction	125
8.2 System Architecture	128
8.3 Adjusting Fuzzy Membership Functions Through Inductive Learning	130
8.4 Learning Fault Diagnosis Heuristic Rules	132
8.5 Case Studies	133
8.6 Conclusions	135

<b>9. Generating Fault Detection Heuristic Rules Through Shallow and Deep</b>	
<b>Knowledge of the Process</b>	<b>137</b>
9.1 Introduction	137
9.2 Description of System Structures	140
9.3 A Methodology for Generating Fault Detection Heuristic Rules	142
9.4 Formulation of Fault Detection Rules for the Mixing Process	143
9.5 Formulation of Fault Detection Rules for the CSTR	148
9.6 Conclusions	152
<b>10. On-Line Fault Detection and Diagnosis Using a Knowledge Based System</b>	
<b>Coupled with a Fuzzy Neural Network</b>	<b>154</b>
10.1 Introduction	154
10.2 System Architecture	156
10.3 Artificial Neural Networks	160
10.4 Fuzzy Neural Networks	168
10.5 Backpropagation Training	173
10.6 Fault Detection and Diagnosis System Applied to the Mixing Process	183
10.7 Fault Detection and Diagnosis System Applied to the CSTR	189
10.8 Performance of the Fault Detection and Diagnosis System	193
10.9 Conclusions	194
<b>11. Conclusions and Recommendations for Future Research</b>	<b>196</b>
<b>References</b>	<b>202</b>

# List of Tables

5.1	Model parameters and nominal values of certain process variables	71
5.2	Controllers parameters	79
6.1	Addition of fuzzy numbers	90
6.2	Subtraction of fuzzy numbers	90
6.3	Predicted behaviour of process variables under "Hand valve 1 is blocked" fault situation	92
6.4	Predicted behaviour of process variables under "Hand valve 2 is blocked" fault situation	93
6.5	Single faults representation for "The Mixing Process"	97
6.6	Fuzzification parameters for "The Mixing Process"	99
7.1	Double simultaneous faults representation for "The Mixing Process"	112
7.2	Single faults representation for the "Continuous Stirred Tank Reactor"	115
7.3	Fuzzification values of the CSTR process variables	116
7.4	Double simultaneous faults representation for the CSTR process	119

# List of Figures

2.1	General structure of a production system	26
2.2	Forward chaining procedure	35
2.3	Backward chaining procedure	35
3.1	Membership function of a normal convex fuzzy number	42
3.2	Fuzzy quantity space represented by normal convex fuzzy numbers	45
3.3	Quantity space represented by crisp intervals	46
3.4	Parametric representation of a normal convex fuzzy number	49
4.1	The mixing process	52
4.2	Controller response when a step in the setpoint of temperature is performed	63
4.3	Controller response when a step in the setpoint of level is performed	64
5.1	Continuous stirred tank reactor	67
5.2	Elementary diagram of the CSTR system	68
5.3	Level control loop	78
5.4	Temperature control loop	78
5.5	Recycled flow rate control loop	79
5.6	Controllers performance	80
6.1	General scheme of a computer assisted fault diagnosis system	86
6.2	The representation of the fuzzy quantity space	87
6.3	Changes in the level of tank 1	88
6.4	Fault simulation	98
6.5	Diagnostic of "Hot water control valve fails low" fault	99
7.1	General scheme of a computer assisted multi-faults diagnosis system	105



7.2	First level of the interface man/machine in order to start a fault simulation situation	107
7.3	Interface man/machine to start the simulation of double simultaneous faults in the "Continuous stirred tank reactor"	108
7.4	Fault diagnosis system organisation	109
7.5	Diagnostic of "Hot water control valve fails low AND Hand valve 1 is blocked" faults	113
7.6	Disturbance effects after faults, "Hot water control valve fails low AND Hand valve 1 is blocked", have been occurred	113
7.7	Diagnostic of "Pipe 1 is blocked" fault	118
7.8	Diagnostic of "Pipe 1 is blocked AND Control valve 3 fails high" faults	122
8.1	General scheme of a computer assisted fault diagnosis system with self-learning abilities	129
8.2	Interface man/machine when the diagnosis system fails	130
8.3	Interface man/machine when a new fault or faults occur in the process	132
8.4	Learning the adjustment of fuzzy membership functions	134
8.5	Self-Learning single abrupt fault symptoms in the mixing process	134
8.6	Self-Learning double simultaneous abrupt faults symptoms in the CSTR	135
9.1	A directed graph	140
9.2	The mixing process directed graph	143
9.3	Continuous stirred tank reactor directed graph	148
10.1	Architecture of a on-line fault detection and diagnosis system	158
10.2	Faults simulation main menu	159
10.3	Man/machine interface for incipient fault simulation	160
10.4	Basic processing element	163
10.5	A simple artificial neural network architecture	164

10.6	Single layer perceptron	165
10.7	General topology of a multilayer perceptron neural network	167
10.8	Fuzzy neural network topology for fault diagnosis	170
10.9	Complement sigmoid function	172
10.10	Sigmoid function	172
10.11	Gaussian function	172
10.12	FNN output errors for the mixing process (Derivative offset = 0)	180
10.13	FNN output errors for the mixing process (Derivative offset = 0.19)	180
10.14	FNN output errors for the mixing process (Derivative offset = 0.2)	181
10.15	FNN output errors for the mixing process (Derivative offset = 0.21)	181
10.16	FNN output errors for the CSTR (Derivative offset = 0)	182
10.17	FNN output errors for the CSTR (Derivative offset = 0.13)	183
10.18	Single abrupt fault detection in the mixing process	185
10.19	Double simultaneous abrupt fault detection in the mixing process	186
10.20	Incipient single fault detection in the mixing process	188
10.21	Incipient double simultaneous fault detection in the mixing process	188
10.22	Single abrupt fault detection in the CSTR	190
10.23	Double simultaneous abrupt fault detection in the CSTR	191
10.24	Incipient single fault detection in the CSTR	192
10.25	Incipient double simultaneous fault detection in the CSTR	192



# Acknowledgements

My gratitude goes, first and for most, to my supervisor Professor Peter D. Roberts for his invaluable advice, guidance, encouragement and his unquestionable support throughout my research work.

I would like to express my sincere and deep gratitude to Professor Sá da Costa, "Instituto Superior Técnico", Technical University of Lisbon, for his resourceful encouragement and moral support.

I am also indebted to the Portuguese institutions "Escola Náutica Infante D. Henrique" and "IDMEC (Pólo I.S.T.)" for the support given during the period of my work at City University.

At last, but not least, I am infinitely indebted to my wife Helena, my parents and my parents in law, for their unlimited support, understanding and encouragement. Moreover, I would like to express my gratitude to my wife Helena for her patience and tolerance during the period of my studies. No words or deeds can compensate her, and my son Ricardo. I can only offer them my deep affection.

# Declaration

The author grants power of discretion to the City University Librarian to allow this thesis to be copied in whole or in part without further reference to him. This permission covers only single copies made for study purposes, subject to normal conditions of acknowledgement.

# Publications

The following reports and papers, based on the work described in this thesis, have been published or will be published soon.

Artificial Intelligence Techniques Applied to Industrial Process Control, Fault Detection and Diagnosis, J. M. F. Calado and P. D. Roberts (1994), *The City University Research Memorandum*, CEC/JMFC-PDR-132, August.

Multiple Simultaneous Fault Detection and Diagnosis Through Fuzzy Qualitative Simulation, J. M. F. Calado and P. D. Roberts (1995), *The City University Research Memorandum*, CEC/JMFC-PDR-141, June.

An Intelligent On-Line Supervisory Fault Detection and Diagnosis System, J. M. F. Calado and P. D. Roberts (1995), *Pre-prints of the 7th IFAC/IFORS/IMACS Symposium on Large Scale Systems: Theory and Applications*, London, UK, July 11-13, Elsevier Science, Vol. 2, pp917-922.

A Distributed Intelligent Fault Detection and Diagnosis System Based on a Knowledge Based Approach Coupled with a Fuzzy Neural Network, J. M. F. Calado and P. D. Roberts (1995), *The City University Research Memorandum*, CEC/JMFC-PDR-143, November.

Fuzzy Neural Network Applied to On-Line Fault Detection and Diagnosis of Abrupt and Incipient Faults, J. M. F. Calado and P. D. Roberts (1996), *3rd InstMC/IEE/IEEE (UK and Ireland) Colloquium on Advances in Fault Diagnosis in Process Control*, York, UK, April 22.

Generating Fault Detection Heuristic Rules Through Deep and Shallow Knowledge of the Process, J. M. F. Calado and P. D. Roberts (1996), to be presented on *UKACC/IEE International Conference on CONTROL'96*, Exeter, UK, September 2-5.

On-Line Fault Diagnosis Based on a Fuzzy Neural Network, J. M. F. Calado and P. D. Roberts (1996), to be presented on *2nd Portuguese Conference on Automatic Control*, Porto, Portugal, September 11-13.

The following paper, based on the work described in this thesis, have been submitted for publication.

On-Line Multiple Fault Diagnosis Based on Fuzzy Qualitative Process Models, J. M. F. Calado and P. D. Roberts (1995), submitted to *IEEE Transactions on Systems, Man and Cybernetics*.



# Abstract

In this research the application of artificial intelligence techniques for on-line process control and fault detection and diagnosis are investigated. The majority of the research is on using artificial intelligence techniques in on-line fault detection and diagnosis of industrial processes. Several on-line approaches, including a rule based controller and several fault detection and diagnosis systems, have been developed and implemented and are described throughout this thesis. The research results obtained demonstrate that rule based controllers can be an alternative in situations where conventional mathematical modelling fails to give a high level of automation. The research on on-line fault detection and diagnosis emphasises the use of deep knowledge based approaches. Therefore, two on-line fault detection and diagnosis systems based on qualitative modelling have been implemented. For the first one only single abrupt faults have been considered while the second one can cope with single and multiple simultaneous abrupt faults. In order to overcome the problems associated with the inherent ambiguity of qualitative reasoning, a fuzzy qualitative simulation algorithm, which allows a semiquantitative extension to qualitative simulation, has been investigated. The adoption of fuzzy sets allows a more detailed description of physical variables, through an arbitrary, but finite, discretisation of the quantity space, and also allows common-sense knowledge to be represented through the use of graded membership. Further research concerning self-reasoning has been done for qualitative model based diagnosis approaches. A self-learning system which can find any inappropriate settings of fault detection and diagnosis parameters and also learn fault symptoms from on-line sampled data, has been developed. Through machine learning techniques, the system can adjust fuzzy membership functions of the process variables automatically, as well as build the knowledge base on-line very efficiently. In order to cope with incipient faults and transient behaviour of the process under concern, a distributed on-line fault detection and diagnosis system, consisting of a knowledge based approach coupled with a fuzzy neural network, has been developed. The fault detection task is performed through the knowledge based approach. A systematic methodology for formulating fault detection heuristic rules from knowledge of system structure and component functions has been investigated. Since structural decomposition corresponds to plant topology, such a method could be easier to implement. A fuzzy neural network approach has been used for fault diagnosis. This system combines the advantages of both fuzzy reasoning and neural networks. In order to speed up the fuzzy neural network training task, an extension of the classical backpropagation learning algorithm is also investigated. The research results achieved with this fault detection and diagnosis system reveal a very good performance and reliability provided that the training data is available.



# Chapter 1

## Introduction

### 1.1 Motivation

Artificial Intelligence has gone in and out of fashion several times over the last 30 years. However, nowadays no one can seriously doubt that computers presents us a unique tool for emulating and extending human cognitive abilities. Neither can anyone doubt that, in the twenty first century, human knowledge will be one of the world's most important commodities. As with any difficult enterprise, artificial intelligence has attracted more spectators than players. Scepticism has always been high, just as it was in the early days of aviation. However, the 1980s have seen the beginning of change in this situation, as prototype systems from university laboratories have been developed and subsequently deployed in commercial application. Perfecting a technology, and making it accessible to all, usually takes longer than doing the original R & D.

A major goal of intelligent control systems is to achieve high performance with increased reliability, availability, and automation of maintenance procedures. In many applications, increased requirements on productivity and performance lead to plants operating near their design limits for much of the time. This may often result in system failures, which are typically characterised by critical changes in the inherent dynamics of the system. System failures can potentially result not only in the loss of productivity but also in the loss of expensive equipment and, ultimately, of human lives. In response to these concerns, tighter safety and reliability specifications have been imposed, which has resulted in increased activity on research dealing with on-line fault detection and diagnosis systems for industrial applications. This development was mainly stimulated by the trend of automation towards more complexity and the growing demand of higher control systems availability and security.



The development of real time fault detection and diagnosis systems is fast becoming an issue of primary significance in the design of intelligent and autonomous control systems (Antsaklis and Passino 1992, Stengel 1993, White and Sofge 1993), since it provides the prerequisites for increased reliability, safety, system availability, minimisation of maintenance activities and costs, and the enhancement of system performance via early detection and diagnosis of faults. Early indication concerning which fault or faults are developing can help avoid systems breakdown, mission abortion and catastrophes. Hence, the architectures of fault detection and diagnosis systems are attracting a lot of attention in a wide range of system engineering applications, such as aircraft and propulsion systems, power plants, nuclear reactors, high speed conveyors, and chemical processes.

The reasons mentioned above have motivated the current studies on the application of artificial intelligence techniques to process control engineering. This field includes expertise of process operators related to the operation of a specific process and expertise of control engineers in designing and utilising different control structures and control algorithms. By making full use of such expertise and knowledge, huge economic profit can result. Good controller performance could lead to good product quality, while good supervisory control could reduce energy and raw material consumption. Earlier detection and diagnosis of faults could reduce damage to process equipment's and products and reduce the shut down times of the process and, hence, reduce profit losses. The aim of applying artificial intelligence techniques in process control engineering is to make full use of available expertise and knowledge in order to achieve economical advantages.

Therefore, a rule based controller and several on-line fault detection and diagnosis systems have been developed and implemented during this research, which are briefly introduced in the following sub chapters. The aims of the research carried out on using a rule based controller in on-line control are to provide and investigate control methods for situations where conventional control techniques has not proven to be efficient, instead of replacing the traditional methods in every situation. The aim of the research on using artificial intelligence techniques in on-line fault detection and diagnosis is to explore more systematic and efficient approaches for building real time fault diagnosis systems.

This chapter is organised as follows. In sub chapter 1.2 a rule based controller is presented. Fuzzy qualitative modelling applied for diagnosis of single and double simultaneous abrupt faults, is introduced in sub chapter 1.3. In sub chapter 1.4, the use of machine learning techniques is explored in order to provide fault detection and diagnosis systems with self-learning abilities. In sub chapter 1.5 a distributed intelligent fault detection and diagnosis system, which is based on a knowledge based approach coupled with a fuzzy neural network, is introduced.



## **1.2 A Rule Based Controller**

Traditional control algorithms depend on numerical models of the processes to be controlled. However, it may be difficult to obtain the numerical models for some processes. As a matter of fact, there are many industrial examples where the high-level of supervisory, control and optimisation of manufacturing processes is not automated but is conducted by a combination of process experts and plant operators. In such situations the operators may have a mental model, in a symbolic form, about the process being controlled, and derive control actions from this symbolic model. Artificial intelligence techniques provide a means for dealing with symbolic computation and, hence, based upon these techniques it is possible to develop a program which can handle symbolic models and decide control actions based on such models.

The first system based on artificial intelligence techniques developed in this research is a rule based controller for a simulated mixing process. The rule based controller is designed based on the causal relations inside the process being controlled. These causal relations form a symbolic model of the process, and the control actions are inferred from samples of both controlled and non-controlled process variables through such a symbolic model.

A similar approach has been reported by Zhang, Roberts and Ellis (1988). The rule based controller developed here shares some of their ideas but, instead of describing the process variables behaviour in linguistic terms, in the present approach the behaviour of the process variables is represented through its qualitative values. The aim is to reduce the quantitative precision of the behaviour description but retain the important distinction. Therefore, +, 0 and - are defined as the quantity space, where these symbols represent the cases that a variable is increasing, unchanging and decreasing respectively. Details of the rule based controller proposed here are presented in chapter 4..

## **1.3 Fault Diagnosis Based on Fuzzy Qualitative Modelling**

In the industrial process control domain, process models are sometimes available and, hence, this knowledge can be used for fault diagnosis purposes. However, for some processes, accurate model parameters may not be available and, in some cases, accurate or direct measurements of some process variables may also be unavailable. Therefore, there is a motivation in this context, to avoid the effort and expense of creating, maintaining and computing with rigorous dynamic mathematical models, by focusing on qualitative indicators of process conditions (Oyeleye and Kramer 1988).



The aim of qualitative modelling techniques is to model the process under consideration qualitatively, and the qualitative behaviour of the process, such as the changes that occur in the process variables can be predicted through qualitative simulation. However, previous methods of qualitative modelling have tended to suffer from excessive generation of multiple solutions, which could lead to a loss in diagnostic resolution if these methods were used in practice.

Therefore, several strategies for reducing spurious solutions and/or ambiguity have been pursued. In order to achieve such a goal the qualitative modelling approach proposed here for on-line fault detection and diagnosis utilises the theory of fuzzy sets to give an arbitrary, but finite, discretisation of the representation of process variables. Linguistic variables are defined and interpreted as verbal probabilities and their semantics are represented by fuzzy numbers. Moreover, the adoption of fuzzy sets allows common-sense knowledge to be incorporated in the interpretation of values through the use of membership grades. Following this procedure, the qualitative modelling approach also allows both magnitude and sign information on the functional relationship holding against two or more variables to be represented, resulting in a considerable reduction of the inherent ambiguity of qualitative computation.

In this research, a fuzzy qualitative method, which is based on de Kleer and Brown's (1984) confluence based qualitative physics, is investigated. By this means, changes in the process variables are predicted in terms of fuzzy numbers. When a fault occurs in the process under concern, the actual behaviour of the process variables will deviate from the predicted one and this can be used to detect the occurrence of a fault or faults in the process under consideration. Once a fault or faults are detected a knowledge based fault diagnosis system is triggered in order to locate the hypothetical fault or faults. The diagnosis task is carried out through the comparison between the process variables real behaviour and the fault symptoms stored in the knowledge base. Such a comparison is performed by using fuzzy sets for describing the process variables real behaviour, as well as for representing the fault symptoms.

However, isolating and identifying a process malfunction can be especially difficult in large fault hypothesis spaces such as in the application of fault detection and diagnosis approaches in large scale systems. Therefore, in order to alleviate this problem and enhance the performance and reliability of the knowledge based fault diagnosis system, the current approach has been structured as a production system where the inference engine possesses forward and backward chaining abilities. Thus, following a forward chaining strategy and based on the patterns of violation in the fuzzy qualitative model, a general analysis is performed for finding what faults are possible and, hence, generate a reduced set of fault candidates. Afterwards, through backward chaining and following a procedure of hypothesis formulation and test, a diagnosis is selected and a more detailed analysis is carried out in order to prove or deny the diagnosis using known facts and other



rules. This methodology has the advantage of first indicating where to look for a solution rather than spending a lot of time retrieving unimportant data.

The on-line fault detection and diagnosis system proposed here, which uses fuzzy qualitative modelling coupled with a knowledge based system, has been implemented with two different configurations. The first one, which is described in chapter 6., has only single abrupt fault detection and diagnosis abilities. In chapter 7. an extension of the first approach is considered which can cope with single and multiple simultaneous abrupt faults. Both systems have been successfully applied in simulation studies conducted with a mixing process and with a continuous stirred tank reactor. Moreover, for both on-line fault detection and diagnosis approaches, respectively described in chapters 6. and 7., a self-learning module has been developed which is briefly introduced in the following sub chapter.

## **1.4 Self-Learning Fault Diagnosis Based on Fuzzy Qualitative Modelling**

A new generation of fault detection and diagnosis systems should have the ability to reason their own behaviour and to learn from past experience. With such a goal, some investigations have been performed in building self-reasoning fault detection and diagnosis systems and a self-learning module has been developed. Such a self-learning system is based on the fault detection and diagnosis systems, using qualitative simulation, described in chapters 6. and 7..

The self-learning approach is based on a hybrid inductive and deductive learning approach. When the fault diagnosis system fails to give the location of a detected fault or faults in the process under concern, the self-learning system is triggered. Therefore, the aim is to identify why the fault diagnosis system failed to give the desired diagnosis and to learn from this experience in order to avoid future similar situations.

The performance of the on-line fault detection and diagnosis approaches mentioned above is affected by some threshold values used for firing the qualitative simulation process in order to detect a hypothetical fault. Any inappropriate settings of these parameters could result in a wrong diagnosis or miss a fault. Therefore, it is desirable that a fault diagnosis system can reason its own behaviour and find out any inappropriate parameters when it failed to give a desired diagnosis result. A inductive learning technique is used to backward trace the fault detection and diagnosis reasoning in order to find any parameters which are responsible for not giving the desired output. Once these parameters are identified, new values are evaluated in order to avoid a future similar situation.



The deductive learning ability of the self-learning fault detection and diagnosis system has been used to acquire new knowledge about fault symptoms. When a fault or faults occur at the first time and, hence, the fault diagnosis system failed to locate such fault or faults, the self-learning system starts to investigate the real behaviour process variables, according to the diagnosis reasoning procedure previously performed. This strategy provides values in a linguistic form whose semantics are represented by fuzzy numbers and stored in the knowledge base in order to be used for a possible future diagnosis. This procedure allows us to build the knowledge base on-line, simplifying the knowledge acquisition procedure, as well as the task of introducing all the fault symptoms in the knowledge base.

Simulation studies conducted with a mixing process and with a continuous stirred tank reactor, where single and multiple simultaneous abrupt faults have been considered, have demonstrated a good performance of the self-learning fault detection and diagnosis system. The results achieved, as well as the system itself, are presented in chapter 8..

## **1.5 Process fault diagnosis using a knowledge based system coupled with a fuzzy neural network**

As mentioned in further chapters, dealing with incipient faulty scenarios, where faults evolve gradually instead of suddenly, is a major limitation of the techniques used in the conception of the fault detection and diagnosis systems previously introduced. Therefore, since incipient faults occur frequently in real applications of fault detection and diagnosis systems for industrial processes, investigations have been conducted for developing on-line fault detection and diagnosis system with the capabilities of coping with such a kind of faulty situation. It has been observed that artificial neural networks are a powerful tool for developing fault diagnosis systems which can cope with incipient faults.

The structure of artificial neural networks is based on our understanding of the biological nervous system. Several models have been proposed where all of them attempt to achieve good performance via dense interconnection of simple computational elements. Instead of performing a program of instructions sequentially, artificial neural networks models explore many competing hypotheses simultaneously using massively parallel networks composed of many computational elements connected by links with variable weights. For modelling a specific problem, these weights are adjusted during the artificial neural network learning procedure. In general, a neural network is presented with a training set consisting of a group of examples from which the artificial neural network can learn. These examples, known as training patterns, are represented as vectors. In



diagnostics, a pattern (symptoms) acts as an antecedent from which we can infer a classification (diagnosis).

Moreover, like brains artificial neural networks have the ability to recognise patterns we cannot even define. This property is known as recognition without definition which enables systems to generalise. Since an artificial neural network has the ability to generalise on the tasks for which it is trained, fault diagnosis seems to be a promising field for their application. From this point of view, the ability to generalise may enable the artificial neural network to provide the correct answer when presented with a new input pattern that is different from any pattern used during the training procedure. Hence, once the artificial neural network has been trained with symptoms of abrupt faults, it may be able to classify the same faults under an incipient faulty scenario, where the fault symptoms will be slightly different from the fault symptoms used during the training task. Moreover, since artificial neural networks can be trained to have the required relationships between inputs and outputs, they can be used to model systems with high nonlinearity and a wide dynamic operating range.

Therefore, a growing number of fault detection and diagnosis systems based on artificial neural networks have been introduced in the last few years. Some of these approaches are mentioned in chapter 10. However, most publications only deal with processes under steady-state conditions or simply are turned off when a transient behaviour occurs. For such systems, and under transient behaviours of the process under consideration, the change in the artificial neural network inputs can also affect certain features of the neural network outputs and, hence, the on-line fault detection and diagnosis system could give incorrect information about a fault or faults in the process in the presence of transient behaviours. This reason motivated the development of a distributed on-line fault detection and diagnosis system, which has been designed to cope with transient behaviours of the process under concern.

The overall computational system implemented consists of a knowledge based approach coupled with a fuzzy neural network. Fault detection is performed through the knowledge based system where fault detection heuristic rules have been generated from deep and shallow knowledge of the process under consideration. Deep knowledge is obtained from structural decomposition of the overall process into subsystems according to plant topology. Details of this procedure are described in chapter 9., where a systematic methodology for generating fault detection heuristic rules is proposed. Since the method developed is systematic, it may be suitable for large scale process analysis. Fault detection heuristic rules based on shallow knowledge have been developed from operational process experience.

The fuzzy neural network performs the diagnosis task. The aim of this approach is to combine the advantages of both fuzzy reasoning and neural networks. Fuzzy reasoning is capable



of handling uncertain and imprecise information while an artificial neural network is capable of learning from examples. Moreover, the fuzzy approach also makes the system less sensitive to measurement noise (Zhang and Morris 1994).

The interface between the knowledge based fault detection approach and the fuzzy neural network has been performed through transfer of data. Following this procedure the system can cope with on-line fault detection and diagnosis in the presence of transient behaviours, without the fuzzy neural network outputs being affected by measurement variables transient behaviour. The distributed on-line fault detection and diagnosis system has been implemented through a TURBO C++ program and successfully applied in simulation studies of a mixing process and of a continuous stirred tank reactor. Single and double simultaneous abrupt faults, as well as incipient faults, have been considered. The results achieved during these simulation studies, as well as a description of the overall computational system, are presented in chapter 10..

## **Chapter 2**

# **Knowledge Based Systems**

### **2.1 Introduction**

Artificial Intelligence (AI) has achieved considerable success in the development of knowledge based systems since mid-1960s. This area of AI has concentrated on the construction of high performance programs in specialised professional domains, a pursuit that has encouraged an emphasis on the knowledge that underlies human expertise and has simultaneously decreased the apparent significance of domain independent problem-solving theory. Moreover, in the last few years the number of expert system applications have been increasing dramatically. As a matter of fact, one can find a large number of reported applications in the periodicals and conference proceedings of many subjects. The terms "expert systems" and "knowledge based systems" are used interchangeably in some artificial intelligence literature (Harmon and King 1985). However, in the remainder of this thesis the author will use the term knowledge based systems.

The area of knowledge based systems investigates methods and techniques for constructing systems with specialised problem-solving expertise. Expertise consists of knowledge about a particular domain, understanding of domain problems, and skill at solving some of these problems. Elucidating and reproducing such knowledge is the central task in building expert systems. Knowledge in any speciality is usually divided in two kinds, public and private. Public knowledge includes the published definitions, facts, theories of which textbooks and references in the domain of study are typically composed. However, expertise usually involves more than this public knowledge. Human experts generally possess private knowledge that has not found its way into the published literature. This private knowledge consists largely of rules of thumb that have come to be called heuristics. Several knowledge based systems have been implemented during this research,



which are presented in the following chapters. This chapter provides some background for a better understanding of the further chapters.

In general, a knowledge based system should consist of the following components (Pham 1988):

1. A knowledge base containing knowledge (facts, information, rules of judgement) about a problem domain.
2. An inference mechanism (also known as inference engine, control structure, or reasoning mechanism) for manipulating the stored knowledge to produce solutions to problems.
3. A user interface (or explanation module) to handle communication with the user in natural language.
4. A knowledge acquisition module to assist with the development of the knowledge base.

The knowledge base and inference engine constitute the core of the system and thus are essential parts of it. For this reason, they will be examined in detail below.

In some artificial intelligence literature the term knowledge engineering has been adopted to combine scientific, technological and methodological elements. A principle of knowledge engineering holds that expert performance rarely conforms to some rigorous algorithmic process, yet this performance does lend itself to computerisation. Knowledge engineering addresses the problem of building skilled computer systems, aiming first at extracting the expert's knowledge and then organising it in an effective implementation. There are several ways of representing knowledge. The three most popular of these are rules, frames and semantic nets (Waterman 1985). Rule-based representation is a shallow representation, whereas schemes using frames and semantic nets are deep representations.

In a rule based system, knowledge is represented in terms of facts pertinent to a problem area and rules for manipulating the facts. As in the approaches described in further chapters, many systems also incorporate information about when or how to apply the rules (i. e. meta knowledge). Facts are asserted in statements which explicitly classify objects or specify relationships between them. Rules consist of modular pieces of knowledge in the form "IF antecedent THEN consequence" or "IF situation THEN action", meaning that if the situation described in the antecedent part of the rule is true, then produce the action specified in the consequence part. Hence the names "IF-THEN rules" or "Production rules" are introduced. In contrast, representation schemes using frames or semantic nets allow a deeper insight into underlying concepts and causal



relationships and facilitate the implementation of deeper-level reasoning such as abstraction and analogy.

However, rule based systems are the most widely used. This is, perhaps, because their development has been greatly facilitated by the availability of low cost skills on personal computers. They tend to be the natural choice for deviation-type problems, such as control, diagnosis, interpretation and monitoring (Pham 1988).

As quoted above, another essential part of the system is the inference mechanism. In a rule based system, the inference engine, also called a rule interpreter, examines facts and executes rules contained in the knowledge base according to set logical inference and control procedures. Reasoning by the exercising of inference rules can proceed in different ways according to different control procedures: backward or forward. In backward chaining, the inference engine works backward from a hypothesised consequence to locate known predicates that would provide support. In forward chaining, the inference engine works forward from known predicates to derive as many consequences as possible.

Although the basic ideas of intelligent problem-solving allow for a wide diversity of implementations and, hence, a few architectural principles have begun to emerge. In this context the term architecture refers to the science and method of design that determine the structure of the knowledge based system. The emergent principles reflect current understanding of the best way to design structures that support intelligent problem-solving. During the current research, the knowledge based systems implemented, have been structured in a similar fashion to production systems (Rich and Knight 1991).

This chapter is organised as follow. Sub chapter 2.2 provides a description of production systems. In sub chapter 2.3 the knowledge acquisition issue is discussed. In sub-chapter 2.4 a detailed description of how to represent the acquired knowledge is given. The inference mechanisms, such as forward and backward chaining, of a knowledge based system are described in sub chapter 2.5.

## **2.2 Production Systems**

Knowledge based technologies are emerging as important elements of process engineering. The role of computing in this area has traditionally and predominantly been in "number-crunching" applications (McDowell *et al.* 1991). However, recent advances in knowledge based techniques allow the application of additional problem solving methods that can enhance and augment numerical techniques and in many cases address issues and problems outside the field of activity of

numerical methods (Stephanopoulos 1990). Production systems have been found to be a powerful tool for enhancing performance and reliability of knowledge based systems.

Since search forms the core of the knowledge based systems implemented during this research, it is useful to structure the program in a way that facilitates the search process. The author has used a structure based upon a production system. A number of knowledge based systems have used this format and have shown that it works well, such as MYCIN and Prospector (Vadera 1989). In general, a production system consists of:

1. A set of heuristic or production rules, each consisting of a left side, which is usually denoted by a rule condition, that determines the applicability of the rule, and a right side that describes the action to be performed if the rule is applied;
2. One or more knowledge bases that contain whatever information appropriate for the particular task. Some parts of each knowledge base may be permanent, while other parts of it may pertain only to the solution of the current problem;
3. A control strategy that specifies the order in which the rules will be compared to the data in the knowledge base and a way of resolving the conflicts that arise when several rules match at once.

A general structure of a production system is shown in Figure 2.1.

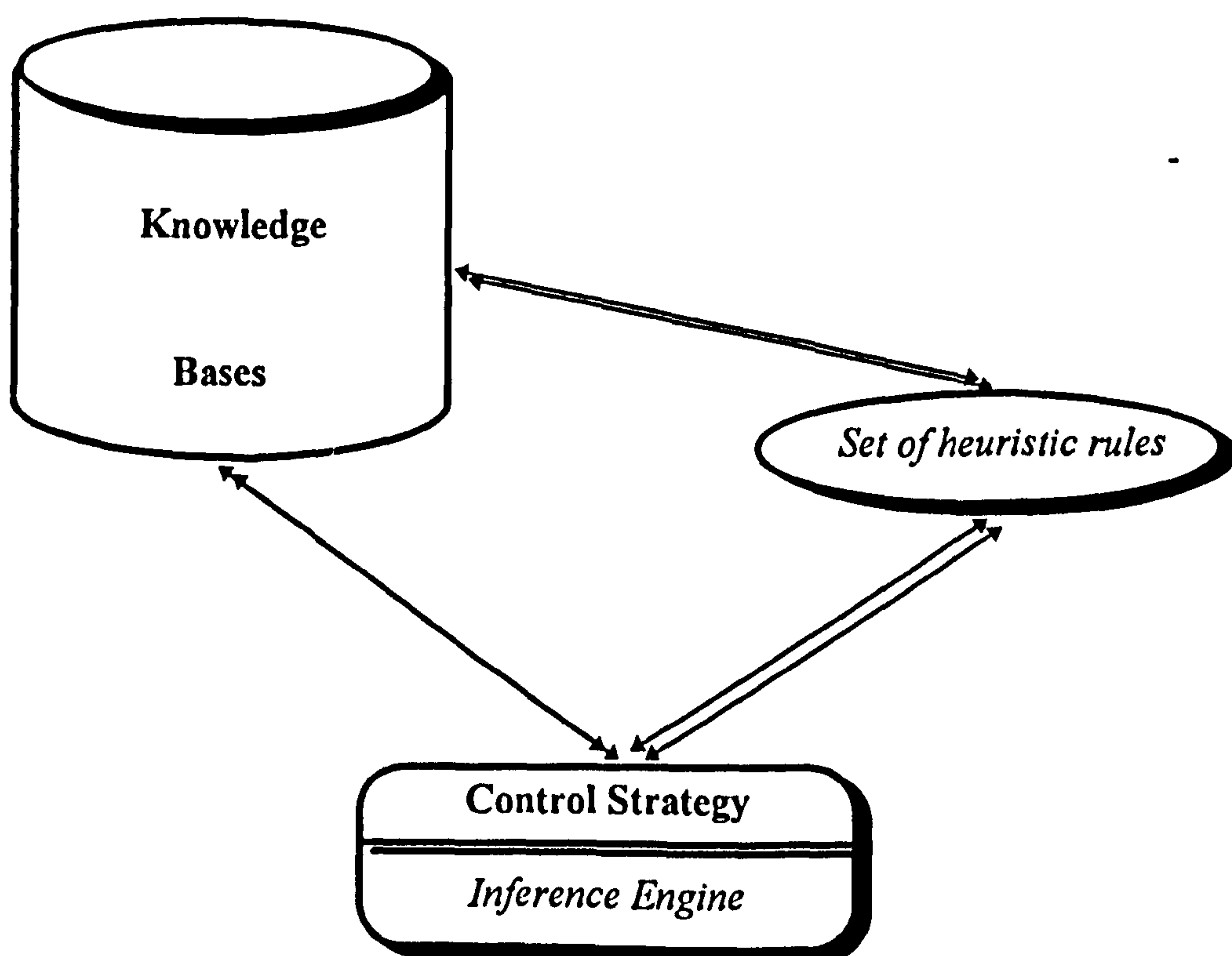


Figure 2.1 - General structure of a production system



The pure production system model has no mechanism for recovering from dead ends in the search task; it simply continues until no more heuristic rules are enabled and then halts. Many practical implementations of production systems allow backtracking to a previous state of the search procedure in such situations.

Production systems provide a model for encoding human expertise in the form of heuristic rules and designing pattern-driven search algorithms, tasks that are central to the design of the rule based knowledge systems. In knowledge based systems, the production system is not necessarily assumed to actually model human problem-solving behaviour; however, the aspects of production systems that make them useful as a potential model of human problem solving make them an ideal tool for building knowledge based systems.

Production systems are a good way to model the strong data-driven nature of intelligent actions. As new inputs enter the database, the behaviour of the system will change. Moreover, new rules can easily be added for new situations without disturbing the rest of the system. Although sometimes confusion arises from interaction amongst rules, it is often less severe than the corresponding complications of modifying straight-line code.

Data-driven search begins with a problem description, such as a set of logical axioms, symptoms of an illness, or a body of data that needs interpretation, and infers new knowledge from the data. This is done by applying rules of inference, legal moves in a game, or other state generating operations to the current description of the world and adding the results to that problem description. This procedure continues until a goal is reached.

Although we have treated production systems in a data-driven fashion, they may also be used to characterise goal-driven search. Goal-driven search begins with a goal and works backwards to establish its truth. To implement this in a production system, the program must try to match the goal against the actions of the heuristic rules. These actions are matched just as the conditions of the heuristic rules were matched in the data-driven reasoning. All heuristic rules whose actions match the goal form the conflict set.

Therefore, the object of a search procedure is to discover a path through a problem space from an initial configuration to a goal state. As mentioned earlier there are two directions in which such a search could proceed:

- Forward, from the start states;
- Backward, from the goal states.

The productions system model of the search process provides an easy way of viewing forward and backward reasoning as a symmetric process. To reason forward, the left sides of the



heuristic rules, the preconditions, are matched against the current state, and the right side, the results, are used to generate new hypotheses until the goal is reached. To reason backward, the right sides are matched against the current hypothesis and left sides are used to generate new hypotheses representing new goal states to be achieved. This continues until one of these goal states is matched by an initial state.

We can also employ a combination of strategies. For example, we can search in a forward direction until the number of states becomes large and then switch to a goal directed search to use possible subgoals to select among alternative states. The danger in this situation is that, when heuristic or best-first search is used, the parts of the chain rules actually searched may "Miss" each other and ultimately require more search than a simpler approach. However, when the branching of a space is constant and exhaustive search is used, a combined strategy can cut back drastically the amount of space searched. For this reason, in the knowledge based fault diagnosis approaches described in subsequent chapters both strategies are combined in order to achieve a good performance for the fault diagnosis system.

Hence, the production system offers a general framework for implementing search. Because of its simplicity, modifiability, and flexibility in applying problem-solving knowledge, production systems have proved to be an important tool for the construction of knowledge based approaches. The production system is an elegant model of separation of knowledge and control in a computer program. Control is provided by an inference engine responsible for firing the heuristic rules, and the problem-solving knowledge is encoded in the rules themselves. The advantages of this separation include ease of modifying the knowledge base without requiring a change in the code for program control and, conversely, the ability to alter the code for program control without changing the set of heuristic rules. The first advantage is particularly important for automating the knowledge acquisition task by using machine learning techniques.

Another important aspect of the production system model is the lack of any syntactic interactions between heuristic rules. Rules may only affect the firing of other heuristic rules by changing the pattern in working memory; they may not "call" another rule directly as if it were a subroutine, nor may they set the value of variables in other heuristic rules. The scope of the variables of these rules is confined to the individual rule. This syntactic independence supports the incremental development of knowledge based systems by successively adding, deleting, or changing the knowledge (heuristic rules) of the system. This property of production systems has been found to be fundamental for implementing knowledge based approaches with self-learning capabilities.

Moreover, the problem addressed by artificial intelligence programs requires particular flexibility in program execution. This goal is served by the fact that the rules in a production



system may fire in any sequence. The description of a problem that makes up the current state of the world determines the conflict set and, consequently, the particular search path and solution.

In artificial intelligence literature it is usual to find the production systems divide into four classes, as monotonic production systems, nonmonotonic production systems, partially commutative production systems and commutative production systems. A monotonic production system is a production system in which the application of a heuristic rule never prevents the later application of another rule that could also have been applied at the time the first rule was selected. A nonmonotonic production system is one in which this is not true. A partially commutative production system is a production system with the property that if the application of a particular sequence of rules transforms state  $x$  into state  $y$ , then any permutation of those rules that is allowable, that is where each rule's preconditions are satisfied when it is applied, also transforms state  $x$  into state  $y$ . A commutative production system is a production system that is both monotonic and partially commutative (Nilsson 1980).

The significance of these classes of production systems lies in the relationship between classes and appropriate implementation strategies. Obviously, for deciding which class of production system is best suited for solving a specific problem, the relationships between the problem and the different types of production systems should be analysed.

Some authors have argued that, for any solvable problem, there exists an infinite number of production systems that describe ways to find solutions. However, some will be more natural and efficient than others. Clearly, any problem that can be solved by any production system can be solved by a commutative one. However, since such a production system may use individual states to represent sequences of applications of rules of a simpler noncommutative production system, hence, it may be practically useless. Therefore, one can conclude that, in a practical sense, there definitely is a relationship between kinds of problems and the kinds of production systems that lend themselves naturally for describing those problems.

For instance, consider the generic problem of classification. The task here is to examine an input and then decide which of a set of known categories the current input is an example. Most diagnosis tasks, including medical diagnosis as well as diagnosis of faults in industrial processes, are examples of classification. For such problems a partially commutative, monotonic production system may be suitable. This procedure will be followed in further chapters for fault diagnosis purposes. From known facts a reduced set of hypothetical faults is generated and then the program using the known facts try to confirm a fault from the reduced set generated.

The methodologies of acquiring knowledge about a specific domain for building the knowledge base of a production system is also a very important subject. Therefore, the next sub chapter examines such methodologies.



## 2.3 Knowledge Acquisition

Knowledge of a domain takes many forms. When that knowledge is firm, fixed and formalised, algorithmic computer programs that solve problems in such a domain are more appropriate than heuristic ones. However, when the knowledge is subjective, ill-codified and partly judgmental, knowledge based systems embodying a heuristic approach are more appropriate. This type of knowledge is rarely formulated in a fashion that permits simple translation into a program. Thus, the process of extracting knowledge from a domain expert and transferring it to a program is an important and difficult problem. This process is denoted by knowledge acquisition and involves problem definition, implementation and refinement, as well as representing facts and relations acquired from an expert.

Knowledge acquisition is a bottleneck in the construction of a knowledge based system. Since the knowledge engineer has far less knowledge of the domain than an expert, the knowledge engineer's job is to act as a go-between to help an expert build a system. However, communication problems impede the process of transferring expertise into a program. The vocabulary initially used by the expert to talk about the domain with a novice is often inadequate for problem-solving; thus, the knowledge engineer and domain expert must work together to extend and refine it. One of the most difficult aspects of the knowledge engineer's task is helping the expert to structure the domain knowledge, to identify and formalise the domain concepts.

Knowledge for a knowledge based system can be acquired in several ways, all of which involve transferring the expertise needed for high performance problem-solving in a domain from a source to a program. The source is generally a human expert but could also be empirical data, case studies, or other sources from which a human expert's own knowledge has been acquired. The process of translating the knowledge from the source to the program may be performed by a knowledge engineer or by a program.

The knowledge used in many of the early knowledge based systems was hand-crafted. A programmer would transform an expert's knowledge into code without separating the knowledge from the reasoning mechanism. Hand-crafting knowledge requires that the programmer learn enough about the domain, but it does not assume that the expert has any knowledge of computing or of the specific implementation involved. The programmer already is an expert or quickly becomes one. It takes a great deal of effort to build and debug such a program, and it is nearly impossible to keep the problem-solving knowledge consistent when frequent updating occurs.

Therefore, since the knowledge to build the knowledge base is acquired partly from past experience, there is reason to hope that a self-learning program could build a knowledge base for a knowledge based system in a similar way. Hence, in a further chapter, a fault detection and



diagnosis system with self-learning abilities, which has the capability of acquiring fault symptoms on-line, has been investigated.

## 2.4 Knowledge Representation

As a research area in its own right, knowledge representation evolved within the field of artificial intelligence, where it continues to play a central role. This should not come as a surprise since everyone would probably agree that intelligence has a lot to do with how knowledge is being handled in the human mind and, hence, in any kind of intelligent device.

Elicited knowledge is recorded in a way that is suitable for people to examine and understand it. However, before it can constitute a knowledge base that is suitable for a computer, it must be organised in such a fashion that a computer inferencing program will be able to access this knowledge whenever needed and draw conclusions.

The function of any representation scheme is to capture the essential features of a problem domain and make that information accessible to a problem-solving procedure. It is obvious that the representation language must allow the programmer to express the knowledge needed for a problem solution. Abstraction, the representation of only that information needed for a given purpose, is an essential tool for managing complexity. Abstraction is also important, in part, because many assumptions and expectations seem to be indexed for storage and retrieval according to the patterns that tie acts together. From the practical point of view, abstraction is also important for conveying information rapidly. This property is specially important when knowledge based systems are applied for real-time fault diagnosis purposes of large scale systems, where a large number of fault symptoms has to be handled.

It is also important that the resulting programs be computationally efficient. Expressiveness and efficiency are major dimensions for evaluating knowledge representation languages. Sometimes, expressiveness must be sacrificed to improve efficiency. Of course, this must be done without limiting the abilities of the representation scheme used to capture essential problem-solving knowledge. Optimising this trade-off is a major task for designers of intelligent systems.

A good system for representation of knowledge in a particular domain should possess the following four properties (Rich and Knight 1991):

1. *Representational Adequacy*, the ability to represent all of the kinds of knowledge that are needed in that domain;



2. *Inferential Adequacy*, the ability to manipulate the representational structures in such a way as to derive new structures corresponding to new knowledge inferred from old;
3. *Inferential Efficiency*, the ability to incorporate into the knowledge structure additional information that can be used to focus the attention of the inference mechanisms in the most promising direction;
4. *Acquisitional Efficiency*, the ability to acquire new information easily. The simplest case involves direct insertion, by a person, of new knowledge into the database. Ideally, the program itself would be able to control knowledge acquisition.

Unfortunately, no single system that optimises all of the capabilities for all kinds of knowledge has yet been found. As a result, multiple techniques for knowledge representation exist and many programs rely on more than one technique. As mentioned above, the three most popular techniques for knowledge representation are semantic networks, frames and heuristic rules.

Semantic networks, also called semantic nets, are basically graphical representations of knowledge that show hierarchical relationships between objects. They are composed of nodes and links between nodes. Each node represents objects and descriptive information about those objects. Objects can be any physical item, concepts, events, or actions. Attributes of an object can also be used as nodes. These might represent size, colour, class, age, origin, or other characteristics. The links between nodes show the relationships between the various objects and descriptive factors. The most common links are of the "is-a" or "has-a" type. Hence, semantic networks are well-suited for representing knowledge of a hierarchical nature. For instance, semantic networks are eminently suitable in recording the way in which objects are made up of their component parts as, for example, in computer-aided manufacturing (Marshall 1990).

A frame is a data structure that includes all the knowledge about a particular object. As a means of representing knowledge, the frame is based on the observation that people do not construct their ideas about familiar objects from scratch, but carry with them a set of expectations about these things. A frame represents an object or situation by describing the collection of attributes that it possesses. It does this by listing all the attributes of a typical case, and by providing a slot for each. This description of the typical case can then be used to capture any individual case by placing the values of its attributes in the respective slots. Frames are basically an application of object-oriented programming for artificial intelligence and knowledge based systems.

Since the knowledge based systems implemented during this research work have followed a production system architecture, the knowledge has been represented through heuristic rules. Following this procedure, knowledge can be represented with heuristic rules of the general form:



- IF antecedent THEN consequence.

According to such a general form, the rules' antecedent, in specific circumstances, can be either true or false, and the rules' consequence will be carried out only when the antecedent is true. The antecedent part of the rule may include dozens of Ifs and the consequence side may include several parts as well. Heuristic rules can be divided into two types such as, declarative rules and procedural rules. Declarative rules state all the facts and relationships about a specific problem. Procedural rules, on the other hand, advise on how to solve a problem, given that certain facts are known.

Heuristic rules representation is especially applicable when there is a need to recommend a course of action based on observable events such as fault diagnosis. It has several major advantages:

- Heuristic rules are easy to understand. Knowledge can be organised into modular form. They are communicable because they are a natural form of knowledge;
- Inference and explanation are easily derived;
- Modifications and maintenance are relatively easy. Knowledge can be added to a knowledge base in a straightforward way by adding heuristic rules to it;
- Uncertainty is easily combined with rules. For instance, in the knowledge based fault diagnosis systems presented in further chapters, fault symptoms are described by using fuzzy sets;
- Each rule is usually independent of all others.

The major limitations, of representing knowledge about a particular domain by using the heuristic rules format, are as follows:

- Complex knowledge requires many heuristic rules;
- Systems with many rules may have a search limitation in the control program. Some programs have difficulty in evaluating rule-based approaches and making inferences.

## 2.5 Inference Engine

Once the knowledge base is completed it is ready for use. To do so, we need a computer program that will enable us to access knowledge for the purpose of making inferences and



decisions. This program is an algorithm that controls some reasoning process and it is usually referred to as the inference engine or the control program. In the heuristic rule based systems it is also referred to as the rule interpreter (Turban 1992).

The inference engine directs the search through the knowledge base. The process may involve the application of procedural rules in what is called pattern matching. The inference engine decides which heuristic rule to investigate, which alternative to eliminate, and which attribute to match. The most popular strategies of an inference engine are forward and backward chaining, which are used in the knowledge based approaches described in further chapters. These inference techniques are described in the remainder of this sub chapter.

Inferencing with heuristic rules involves implementation of the *modus ponens* approach (Rich and Knight 1991). According to this procedure, if there is a heuristic rule such as "IF *A* THEN *B*" and if we know that *A* is true, then it is valid to conclude that *B* is also true. When this situation has occurred, we say that a specific heuristic rule was fired. Firing a rule occurs only when all of the rule's antecedents (IF side) are satisfied.

Testing a rule antecedent or consequence can be as simple as matching a symbolic pattern in the heuristic rule to a similar pattern in the working memory. Every heuristic rule in the knowledge base can be checked to see if its antecedent or consequence can be satisfied by previously made assertions. This process may be done in one of two directions, forward or backward, and it will continue until no more rules can be fired, or until a goal is achieved.

Forward chaining is a data-driven approach. In this approach, we start from available information as it comes in, or from a basic idea, and then try to draw conclusions. Following a forward chaining strategy, the inference engine uses heuristic rules in a chain to move forward from a given knowledge to new knowledge.

For instance, from the fact that *A* is true and the rule, "IF *A* THEN *B*", we may deduce that *B* is true. Afterwards, the existence of a rule such as, "IF *B* THEN *C*", allows us to deduce that *C* is true. This process can be continued for as long as the heuristic rules will connect. Such a chaining procedure may be represented as shown in Figure 2.2. Moreover, forward chaining is equivalent to finding what can be inferred from given knowledge by using a set of rules.

Backward chaining is a goal-driven approach in which we start from an expectation of what is going to happen (hypothesis), and then seek evidence that supports or contradicts our expectation. In other words, we set out to find if *C* is true. A heuristic rule such as, "IF *B* THEN *C*", asserts that *C* is true if *B* is also true. A heuristic rule, "IF *A* THEN *B*", asserts in turn that *B* is true if *A* is also true. Therefore, if *A* is a known fact, then the chain is complete and *C* is shown to be true. Backward chaining is illustrated in Figure 2.3.

The smallness of the set of rules considered above obscures certain difficulties that may

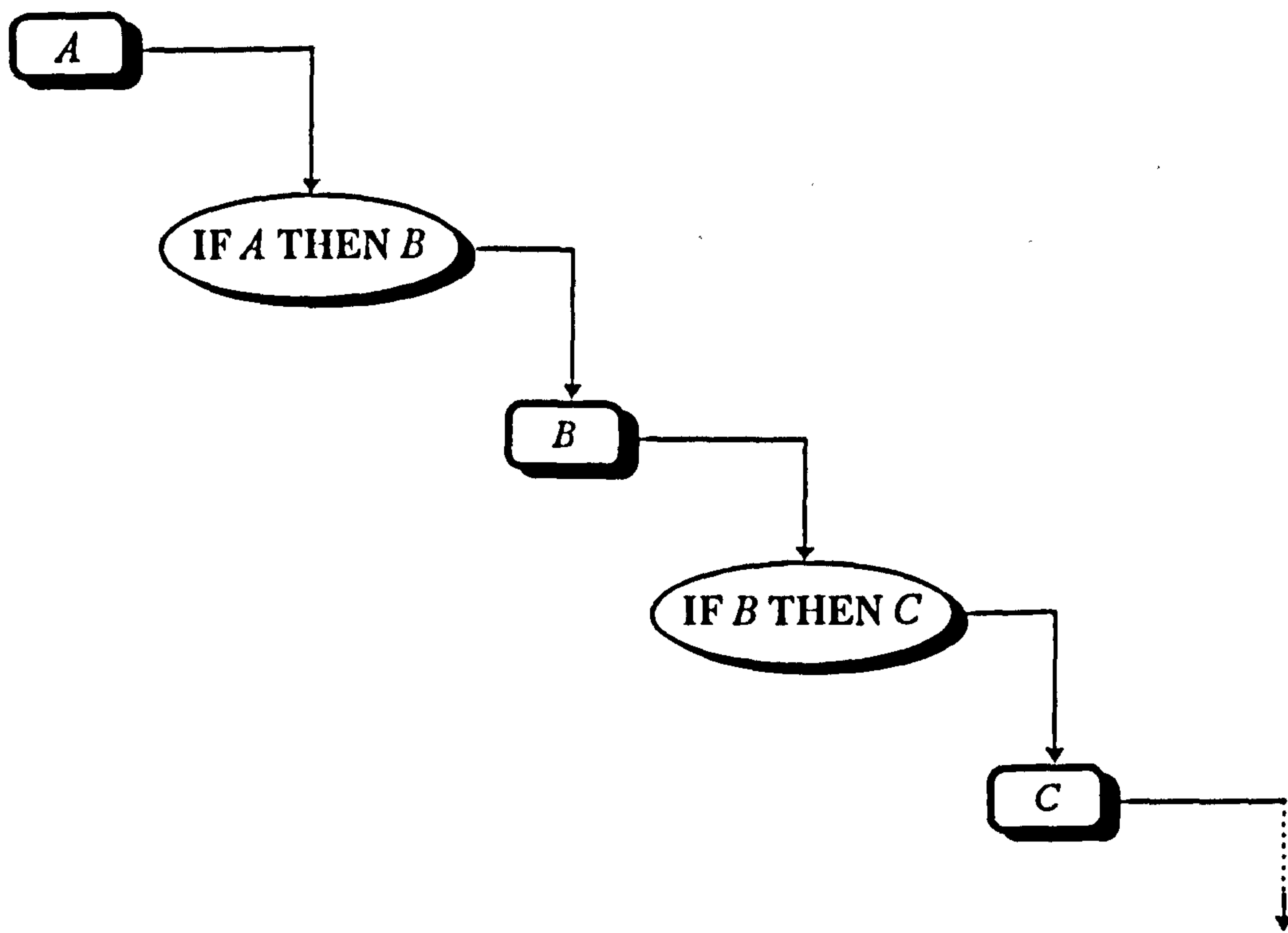


Figure 2.2 - Forward chaining procedure

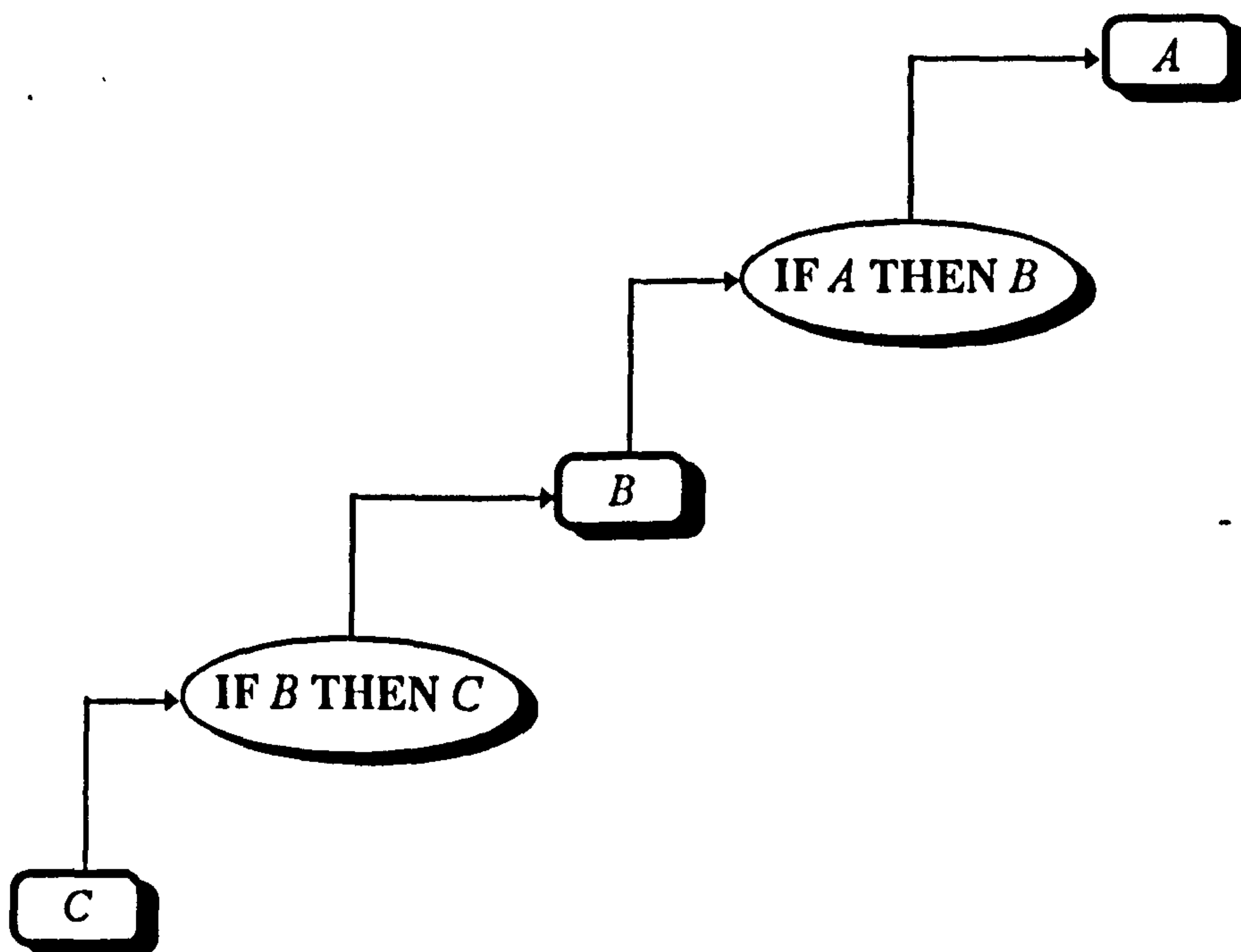


Figure 2.3 - Backward chaining procedure

arise with larger sets. It may be that at any point there is more than one rule that matches the requirement to continue the chain. In this case some means of resolving the conflict must be invoked. The simple expedient of taking the first matching rule to be encountered is employed in some systems, but others employ more sophisticated strategies. Another matter that arises is how to



decide when to use forward chaining and when to use backward chaining. In general, this can be decided by examining the branching factor of the chaining process and deciding which is the smaller of the number of new facts that can be determined from the given ones, and the number of ways in which assertions can be demonstrated to be true in terms of the given facts.

# Chapter 3

## Fuzzy Systems

### 3.1 Introduction

System theory is defined as a body of concepts and techniques used to analyse and design systems regardless of their nature. The important aspect of systems analysis and design is the development of a model that describes the "cause and effect" relationships between variables. But an exact description of any real system is virtually impossible. Our inability to make precise statements about complex behaviours is a fact we have to accept and adjust to. Complexity is associated with description rather than being thought of as an intrinsic property of objects. Hence, we may well consider reducing the complexity of an object, not by changing that object, but by changing our views about it.

Conventional systems approach has tended to assume that what needs to be done is to survey the whole system. The fuzzy systems approach challenges this assumption and proposes to construct a global representation on the basis of assumptions about partial representation. In this way we are able to hold different imperfect representations of reality under concern at the same time. Fuzzy systems theory has already been born and is growing.

This chapter provides the background and key points about fuzzy logic, which form the basis of the work carried out by the author and presented in later chapters. The term fuzzy logic, which has become very popular in the last decade, is used in a generic way throughout this thesis. It is not limited to the narrow view of fuzzy logic, which is a generalisation of conventional logic. Rather it encompasses all methods, techniques and tools based on fuzzy set theory. The term fuzzy technology, which is often used in literature, is probably a better name for the application of fuzzy set theory and we will use it later.



Fuzzy logic was invented in the mid 1960s as an alternative to two-valued logic and probability theory by offering alternatives to traditional notions of set membership and logic (Bellman *et al.* 1964). After years of academic debate on its merits, fuzzy logic has finally emerged as an alternative to classical binary valued logic in applications ranging from industrial process control to consumer products to aerospace and bioengineering (Sugeno 1985, Zimmermann 1991 and Langari *et al.* 1994). Fuzzy logic views two-valued logic and set theory as special cases of a more general multi valued theory. In this approach mathematical probability is viewed as inappropriate and is replaced with an alternative theory referred to as possibility theory (Zadeh 1979).

In contrast to inspection frequencies that tend to use the record of previous events in assigning probabilities, fuzzy truth values deal with the likelihood or certainty that a fact or heuristic rule is true. The main idea behind fuzzy systems is that a truth value (in fuzzy logic) or a membership value (in fuzzy sets) is indicated by a value in the range 0-1; with 0 representing absolute Falsity and 1 representing absolute Truth. Note that in fuzzy set theory, these membership values do not have to sum to 1, in contrast to probabilities that are constrained by a summation axiom. Thus, in fuzzy logic, we may use any form of inexact value assignments without fear of upsetting the underlying mathematical model.

Everyday language is one example of the way vagueness is used and propagated. Imprecision in data and information gathered from and about our environment is either statistical or nonstatistical. This latter type of uncertainty is called fuzziness. Hence, fuzzy sets are intuitively very appealing. Natural language abounds with vague (fuzziness) and imprecise concepts, such as "The temperature is high" or "The water inlet flow is small". Another example of a fuzzy set is the set of real numbers much larger than zero. For this example, the real numbers that are not all larger than zero, are not in the set, while numbers which are larger than zero are partially in the set based on how much larger than zero they are. Thus, the goal behind the introduction of fuzzy set theory was to provide a means of defining categories that are inherently imprecise. Since the introduction of fuzzy set theory the terms hard and crisp have been used to describe sets conforming to traditional set theory.

The theory of fuzzy sets has as its main aim the development of a methodology for the formulation and solution of problems that are too complex or ill-defined to be susceptible to analysis by conventional techniques. Therefore, during the last three decades, the theory of fuzzy sets has developed in a variety of directions, finding applications in solving various kinds of real physical world problems, particularly in the fields of pattern classification, information processing, control, systems identification, artificial intelligence, and, more generally, decision processes involving incomplete or uncertain data.



The advantage provided by fuzzy logic is that the degree of membership in a set can be specified, rather than just the binary is or isn't member. This can be especially advantageous in describing the rate of change of industrial process variables, where frequently objects are not clearly members of one class or another. Using crisp techniques an ambiguous object will be assigned to one class only, lending an aura of precision and definiteness to the assignment that is not warranted. On the other hand, fuzzy logic will specify to what degree the object belongs to each class, which is information that frequently is useful.

This chapter is organised as follows. Sub chapter 3.2 provides some formal definitions for fuzzy sets, which are fundamental for following the reasoning in further chapters. Sub chapter 3.3 presents several alternatives for representing a fuzzy quantity space of a physical system. The problem of performing basic arithmetic operations within a fuzzy quantity space is discussed in the last sub chapter.

## 3.2 Formal Definitions for Fuzzy Sets

Fuzzy sets were introduced by Zadeh (1965) as a new way to represent vagueness in everyday life. This indeterminacy does not mean that one lacks sufficient knowledge of a concept or some facts, but that the concept itself is such that its relation with the current state of affairs is uncertain. It is through man or human thinking that fuzziness comes into the world. In many situations there are several choices but not all are equally acceptable. The conventional quantitative techniques of system analysis are unsuited for dealing with humanistic systems and other comparable complex systems, because, as the complexity increases, our ability to make precise and yet significant statements diminishes until a threshold is reached beyond which precision and significant relevance become mutually exclusive characteristics.

Fuzzy sets are a generalisation of conventional set theory, one of the basic structures underlying computational mathematics and models (Bezdek and Pal 1992). Fuzzy approach is based on the premise that the key elements in human thinking are not just numbers but can be approximated to tables of fuzzy sets, or, in other words, classes of objects in which the transition from membership to nonmembership is gradual rather than abrupt. Computational pattern recognition has played a central role in the development of fuzzy models because fuzzy interpretation of data structures is very natural and intuitive. Fuzzy control theory has also provided a wide variety of real system applications of fuzzy technology. Fuzzy control has emerged as one of the most active fields for research in the application of fuzzy set theory, as well as studies on the theory itself, as have been reported in many works (Sugeno 1985, Lee 1992, Chen



*et al.* 1993, He *et al.* 1993, Yager and Zadeh 1994, Yen *et al.* 1995 and Linkens and Nyongesa 1995).

This sub chapter presents a brief review of the relevant aspects of fuzzy mathematics that forms the basis of the work described in further chapters. A more extensive treatment of fuzzy mathematics can be found in Zadeh *et al.* (1975), Kandel and Lee (1979), Dubois and Prade (1980), Klir and Folger (1988) and Zimmermann (1991).

As mentioned above, the fuzzy sets theory deals with a subset of a universe of discourse, where the transition between full membership of a set and no membership is gradual rather than abrupt. Such subsets, which are called fuzzy sets, arise, for instance, when descriptions of ambiguity, vagueness, and ambivalence in the mathematical models of physical systems are needed. Examples of such a situations are given in the following chapters.

Let us introduce a special notation that is often used in the literature for defining fuzzy sets with a finite support. Let  $X$  be a space of objects, with a generic element of  $X$  denoted by  $x$ . A fuzzy set  $A$  in  $X$  is characterised by a membership function,  $\mu_A(x)$ , which associates each point in  $X$  a real number in the interval  $[0, 1]$ . This membership function can be viewed as a weighting coefficient which reflects the ambiguity in a set. Thus, the nearer the value of  $\mu_A(x)$  to unity, the higher the grade of membership of  $x$  in  $A$ . Moreover, the support of a fuzzy set,  $A$ , in the universal set,  $X$ , is the crisp set that contains all the elements of  $X$  that have a non zero membership grade in  $A$ . This can be expressed as follow,

$$\text{supp } A = \{x \in X \mid \mu_A(x) > 0\} \quad (3.1)$$

A crossover point in  $A$  is an element of  $X$  whose grade of membership in  $A$  is 0.5. A fuzzy singleton is a fuzzy set whose support is a single point in  $X$ . If  $A$  is a fuzzy singleton whose support is the point  $x$ , we may write,

$$A = \frac{\mu}{x} \quad (3.2)$$

where  $\mu$  is the grade of membership of  $x$  in  $A$ . To be consistent with this notation, a nonfuzzy singleton will be denoted by  $1/x$ .

The fuzzy set  $A$  may be viewed as the union of its constituent singletons. Therefore, if we assume that  $x_i$  is an element of the support of fuzzy set,  $A$ , and that  $\mu_i$  is its grade of membership in  $A$ , then  $A$  can be written as follows,

$$A = \mu_1/x_1 + \mu_2/x_2 + \dots + \mu_n/x_n \quad (3.3)$$

Note, that in equation (3.3) the slash is employed to link the elements of the support with their grades of membership in  $A$  and the plus sign indicates, rather than any sort of algebraic addition, that the listed pairs of elements and membership grades collectively form the definition of the set  $A$ .

For the case in which a fuzzy set,  $A$ , is defined on a universal set that is finite and countable, we may express such a fuzzy set through the following equation,

$$A = \sum_{i=1}^n \mu_i/x_i \quad (3.4)$$

In this sense of addition, a finite universe of discourse  $X = \{x_1, x_2, \dots, x_n\}$  may be represented simply by the summation,

$$X = x_1 + x_2 + \dots + x_n \quad (3.5)$$

or

$$X = \sum_{i=1}^n x_i \quad (3.6)$$

although, strictly, we should write the equations (3.5) and (3.6) as follows,

$$X = 1/x_1 + 1/x_2 + \dots + 1/x_n \quad (3.7)$$

and

$$X = \sum_{i=1}^n 1/x_i \quad (3.8)$$

However, when  $X$  is an interval of real numbers, that is when the support of  $A$  is a continuum, a fuzzy set,  $A$ , is often written in the following form,



$$A = \int_x \frac{\mu_A(x)}{x} \quad (3.9)$$

An ordinary set thus becomes a special case of a fuzzy set with a membership function which is reduced to the well known binary value, either 0 or 1, characteristic function. The definitions, theorems, proofs and so on of fuzzy set theory always hold for nonfuzzy sets (Pal and Majumder 1986). Because of this generalisation, the theory of fuzzy sets has a wider scope of applicability than classical set theory in solving problems that involve, to some degree, subjective evaluation.

When  $\mu_A(x)$  is restricted to the values 0 and 1, the fuzzy set,  $A$ , degenerates to an ordinary set and its membership distribution becomes the characteristic function of a classical set. Furthermore, if  $A$  satisfies the two conditions below,

$$a) x \in X, \mu_A(x) = 1; \quad (3.10)$$

$$b) r, s \in X, \lambda \in [0, 1], \mu_A(\lambda r + (1-\lambda)s) \geq \min(\mu_A(r), \mu_A(s)); \quad (3.11)$$

then,  $A$  is a normal and convex fuzzy number (Dubois and Prade 1980). Graphically, the membership distribution of the fuzzy set "approximately  $y$ ", can be represented as shown in Figure 3.1.

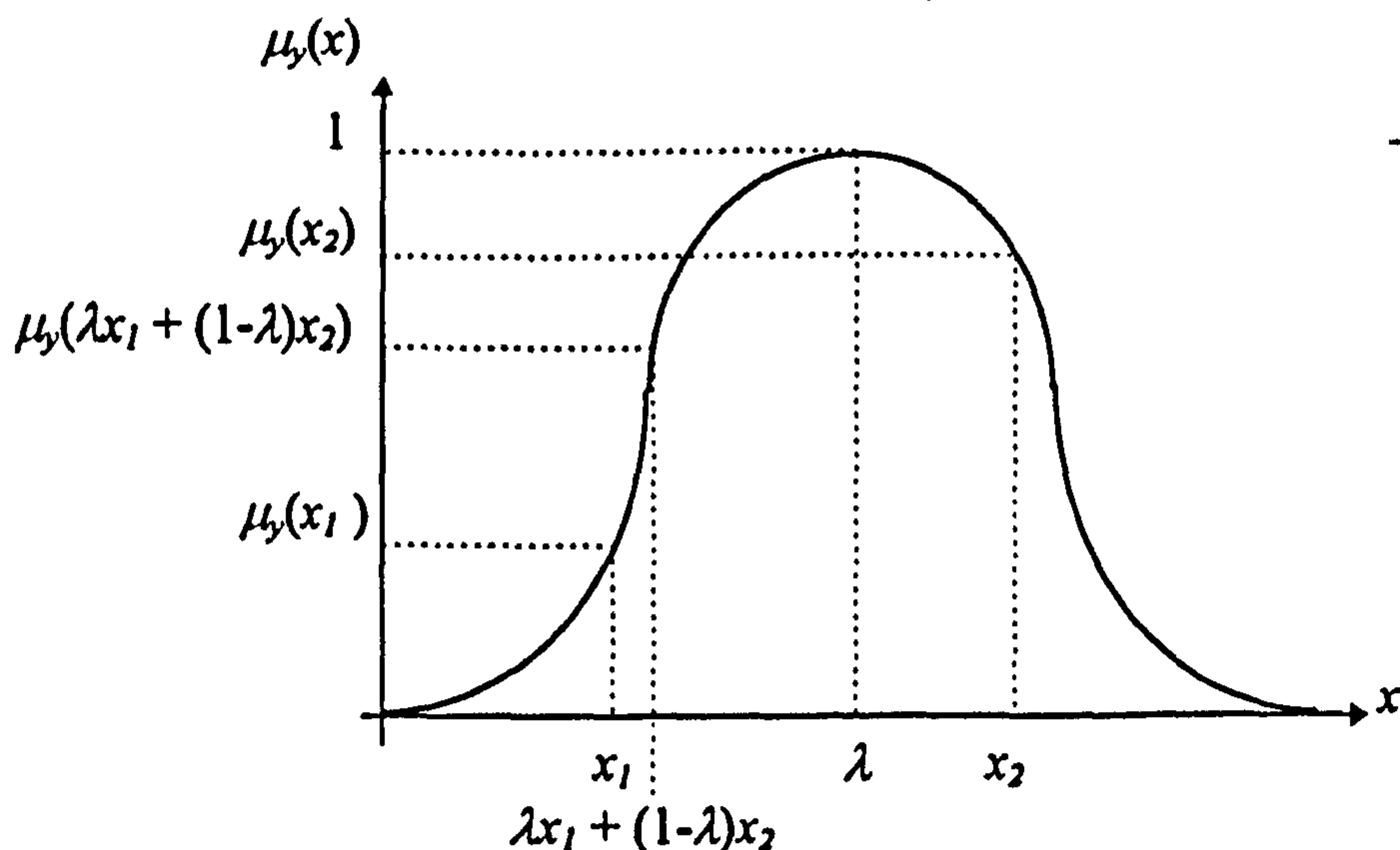


Figure 3.1 - Membership function of a normal convex fuzzy number

The membership values obtained from a specific membership function of a fuzzy set, determine how much fuzziness such a fuzzy set contains. Therefore, because fuzzy sets are a

generalisation of the classical set theory, the embedding of conventional models into a larger setting endows fuzzy models with greater flexibility to capture various aspects of incompleteness or imperfection in whatever information and data are available about a real process. The fuzzy membership function is in some respect similar to the probability density function, however, they are conceptually different. Probability is about how frequently a sample occurs in a population while fuzzy membership value means how closely or how accurately a sample resembles an ideal element of a population.

The assignment of the membership function of a fuzzy set is subjective in nature, and reflects the context in which the problem is viewed. It can not be assigned arbitrarily. For instance, it would be totally wrong to assign the membership function of the fuzzy set "real numbers clustered around 10" using a function which increases monotonically.

When we want to exhibit an element,  $x \in X$ , that usually belongs to a fuzzy set,  $A$ , we may demand its membership value to be greater than some threshold,  $\alpha \in [0, 1]$ . The ordinary set of such elements is called the  $\alpha$ -cut of  $A$  and denoted by  $A_\alpha$  (Zimmermann 1991). Hence, an  $\alpha$ -cut of a fuzzy set,  $A$ , is a crisp set,  $A_\alpha$  that contains all the elements of the universal set,  $X$ , that has a membership grade in  $A$  greater than or equal to the specified value of  $\alpha$ . Such a definition can be written as follows,

$$A_\alpha = \{x \in X \mid \mu_A(x) \geq \alpha\} \quad (3.12)$$

The value  $\alpha$  can be chosen arbitrarily but is often designated at the values of the membership grades appearing in the fuzzy set under consideration.

Using the concept of  $\alpha$ -cut, a fuzzy set  $A$  may be decomposed into its associated  $\alpha$ -cut  $A_\alpha$  through the resolution identity,

$$A = \int_0^1 \alpha A_\alpha \quad (3.13)$$

where the following notation is used:

- $\alpha A_\alpha = \{(x, \alpha) \mid x \in A_\alpha\}$ , stands for a fuzzy set representing the product of a scalar,  $\alpha$ , with the set  $A_\alpha$ ;
- $\int_0^1$  is the union operator on  $\alpha A_\alpha$ , with  $\alpha$  ranging from 0 to 1.

Here, the union of two fuzzy sets is defined by the following expression,



$$A \cup B = \left\{ (x, \mu_{A \cup B}(x)) \mid \mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)), x \in X \right\} \quad (3.14)$$

That is, the resolution identity can be viewed as the result of combining those elements in  $A$  that fall into the same level set.

In the context of fuzzy logic theory, detailed definitions for the operations of union, intersection, complement, algebraic product and algebraic sum were pointed out by Zadeh *et al.* (1975) to investigate fuzzy reasoning. Mizumoto and Tanaka (1978), investigates the algebraic properties of fuzzy sets under such operations.

### 3.3 Fuzzy Quantity Spaces

In later chapters a fuzzy qualitative simulation technique is used for on-line fault detection in industrial processes. Several studies and applications of fault detection and diagnosis systems based on fuzzy logic has been reported (Tsukamoto and Terano 1977, Asse *et al.* 1987, Kitowski and Bargiel 1988, Ulieru and Isermann 1993). All of them describe the process variables behaviour in qualitative terms. However, the choice of representation of physical quantities plays a critical role in qualitative modelling. In the present approaches fuzzy qualitative values are used to provide a semi-quantitative extension to the quantity representation of magnitude of change of a process variable.

All qualitative simulation techniques describe quantities with a small set of symbols, which are called qualitative values. These values are abstracted from the underlying field that the variables of a physical system take values from, sometimes called the support set. In the on-line fault detection approaches described in further chapters, a fuzzy qualitative value of a process variable is a fuzzy number chosen from a subset of normal convex fuzzy numbers. This subset is generated by an arbitrary but finite discretization of the underlying numeric range of the variable. A set, consisting of all the elements of such subsets for all the variables in the process under concern, is called a fuzzy quantity space which is denoted in the remainder of this thesis by  $q_f$ . The real number zero is required to belong to  $q_f$ . It is worth noting that, under the above definition, a variable takes values from a subset of a quantity space  $q_f$ . This subset can be different from the other subsets of  $q_f$  from which other variables take values. Moreover, the rate of change of a variable can also have different sets of qualitative values. Therefore, this increases the flexibility of the representation of knowledge about plants since, if necessary, we can model a physical plant with different detailed abstraction of its variables in response to the extent of we have knowledge of the process variables behaviour.

Because a fuzzy quantity space,  $q_f$ , is generated by a finite discretization of the underlying range of each variable of the process being modelled, we can translate a subset of a numeric range to a single qualitative value according to what is needed in a specific application. Moreover, according to the rule mapping method, a fuzzy quantity space can be represented through the following index set (Peng and Liu 1988, Wang *et al.* 1990, Wong *et al.* 1993),

$$q_f = \{A_m, A_{m+1}, \dots, A_1, A_0, A_1, \dots, A_{m-1}, A_m\} \quad (3.15)$$

The index set, represented by expression (3.15), is denoted as a sequence containing  $2m+1$  linguistic values for a linguistic variable, where a linguistic values set is represented by means of a numerical set. For instance, the following chapters is define a fuzzy quantity space for describing the process variables behaviour, where  $m$  takes the value 3. Therefore, each process variable rate of change is described by seven fuzzy sets,  $A_3, A_2, A_1, A_0, A_1, A_2$  and  $A_3$ , which are called "Negative-Large (*nlarge*)", "Negative-Medium (*nmedium*)", "Negative-Small (*nsmall*)", "Zero (*zero*)", "Positive-Small (*psmall*)", "Positive-Medium (*pmedium*)" and "Positive-Large (*plarge*)". For simplicity, however, a normalised range,  $[-1, +1]$ , for representing the rate of change of each process variable, has been used. This normalised range forms the basis on which the fuzzy quantity space is discretised.

Therefore, by fuzzy representation, the underlying real range  $[-1, +1]$ , from which a process variable takes values, can be mapped onto a set of qualitative values represented by the following fuzzy quantity space, which can be represented as shown in Figure 3.2,

$$q_f = \{nlarge, nmedium, nsample, zero, psmall, pmedium, plarge\} \quad (3.16)$$

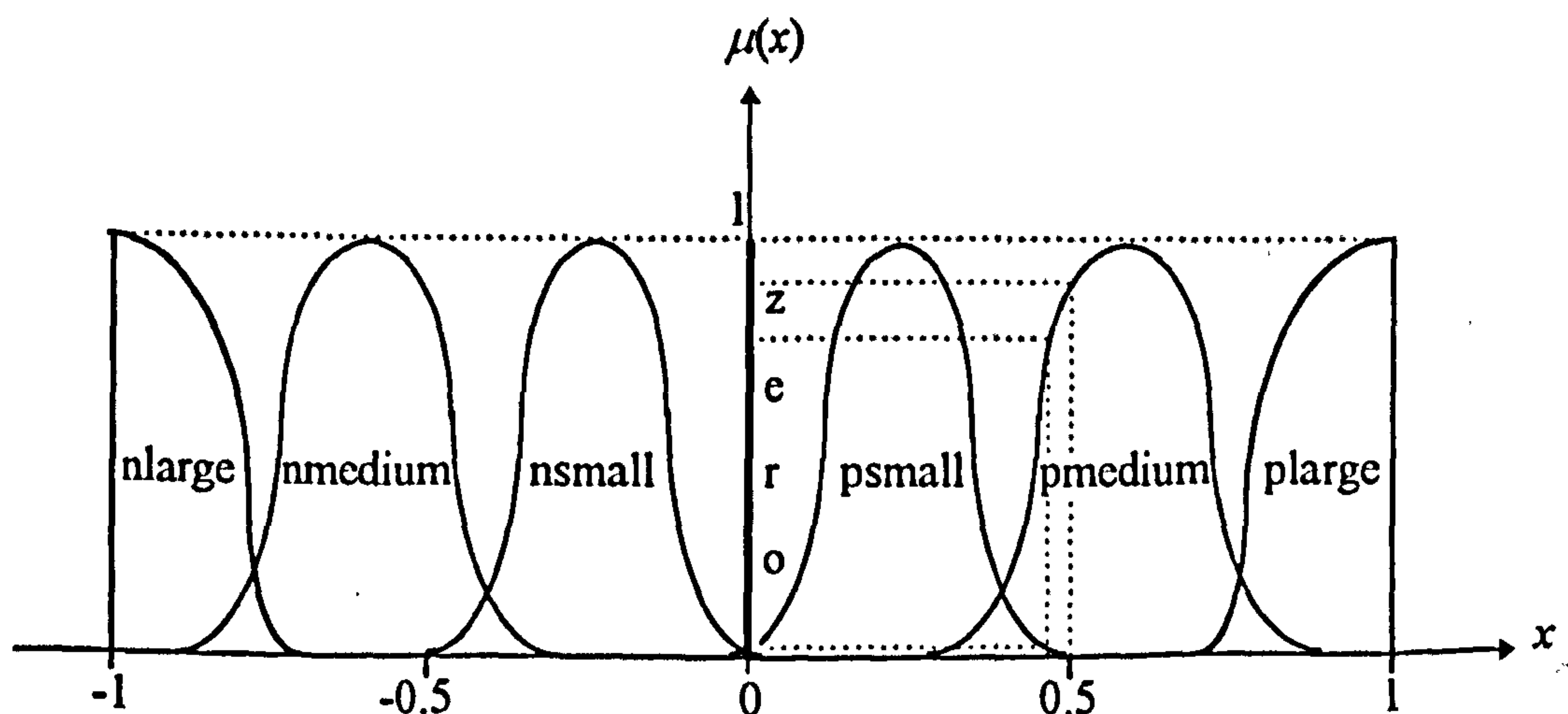


Figure 3.2 - Fuzzy quantity space represented by normal convex fuzzy numbers



Thus, according to Figure 3.2, each qualitative value  $A$ , actually a normal convex fuzzy number, has an associated linguistic term so that it corresponds to the perceived meaning. The fuzzy quantity space removes the boundary interpretations problem, which is achieved through the description of a gradual rather than an abrupt change in the degree of membership of which a physical quantity is mapped onto a particular qualitative value. It is, therefore, closer to our common sense intuition of the description of a qualitative value.

In order to illustrate the boundary interpretation problem, let us consider the following example. According to the fuzzy quantity space depicted in Figure 3.2, a rate of change in a process variable having a value 0.5 belongs to *pmedium* with a membership value less than 1. If such a process variable presents a rate of change equal to 0.45, then it belongs to *pmedium* with a strength less than the previous one, but simultaneously it also belongs to *psmall* with a small membership value. This non-exclusivity of values is an important aspect of fuzzy sets and, again, is important in capturing our common sense intuition. However, using crisp intervals, for instance as shown in Figure 3.3, a rate of change equal to 0.5 belongs to *pmedium* if a close interval is considered while 0.45 does not but fully belongs to the interval defined as *psmall*. Clearly, such a crisp representation would often result in a non-intuitive interpretation in practice.

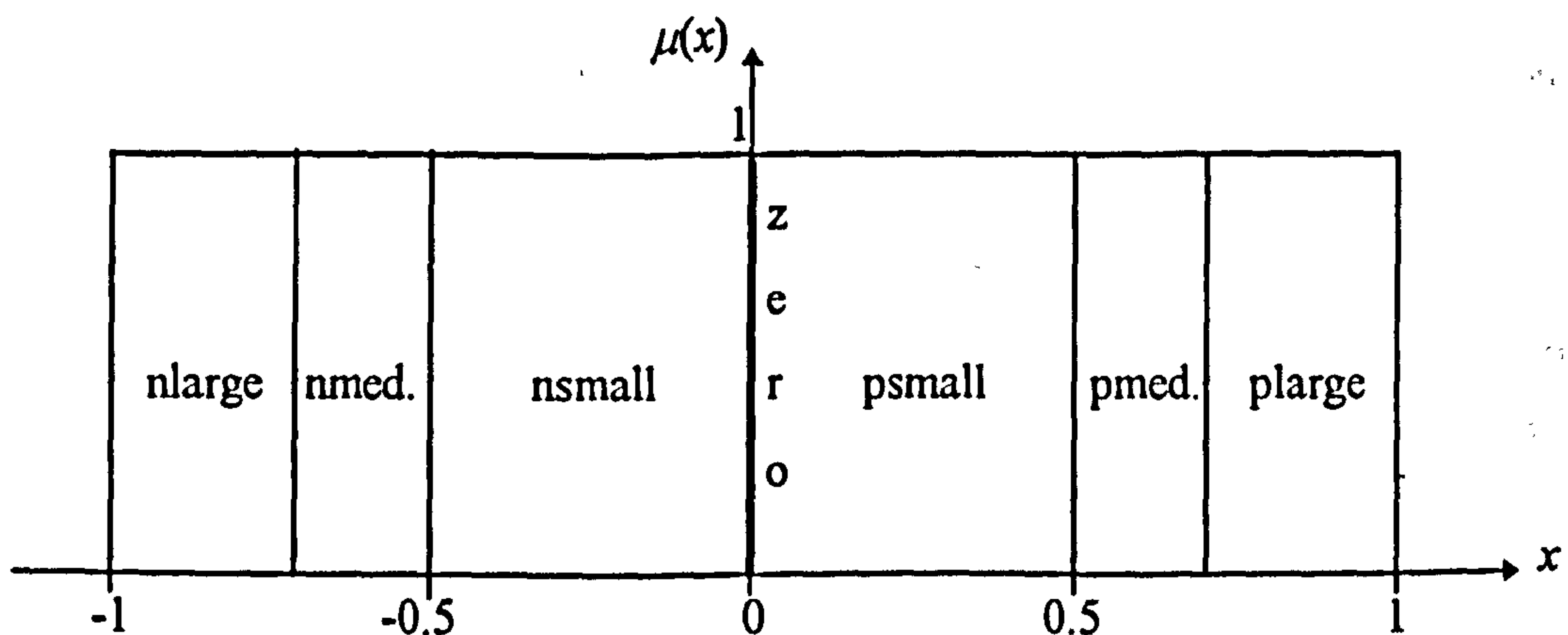


Figure 3.3 - Quantity space represented by crisp intervals

By relying on the use of fuzzy linguistic values, and fuzzy algorithms, this new approach provides an approximate and yet effective and more flexible means for describing the behaviour of systems which are too complex or too ill defined to admit precise mathematical analysis by classical methods and tools.

However, the definition, on a fuzzy quantity space presented above, is given in a general form. It has been assumed that the quantity space consists of normal and convex fuzzy numbers with arbitrary forms of distribution. Moreover, in further chapters, these fuzzy numbers, used for describing the process variables behaviour, will be propagated through a qualitative model of the

process under consideration in order to predict the behaviour of some process variables in fuzzy terms. Arithmetic operations form the basis of this qualitative simulation procedure. However, operations performed within a quantity space consisting of normal convex fuzzy numbers usually entail several types of computational difficulties. As a matter of fact, arithmetic operations on fuzzy qualitative values are based upon the extension principle presented in the next sub chapter. As stated in the next sub chapter, this principle is invoked every time an arithmetic operation is performed and requires expensive calculations. The way to alleviate this problem is presented in the next sub chapter and is followed in further chapters by a fuzzy qualitative simulation algorithm used for fault detection purposes in industrial processes.

### 3.4 The Extension Principle

The extension principle, which has been introduced by Zadeh (1975), provides a general method for extending nonfuzzy mathematical concepts in order to deal with fuzzy quantities. The extension principle can be systematically applied to real algebra, operations of fuzzy numbers, and also for defining set theoretic operations for higher order fuzzy numbers.

The extension principle allows the use of fuzzy sets to represent algebraic operations in a fuzzy framework. It provides the means for any function  $f$  that maps points in  $x_1, x_2, \dots, x_n$  in the crisp set  $X$  to the crisp set  $Y$  to be generalised such that it maps fuzzy subsets of  $X$  to  $Y$ . Formally, given a function  $f$  mapping points in set  $X$  to points in set  $Y$  and any fuzzy set  $A \in P(X)$ , where  $A$  is given by expression (3.9), the extension principle states that  $f(A)$  can be given through the following equation,

$$f(A) = f\left(\frac{\mu_1}{x_1} + \frac{\mu_2}{x_2} + \dots + \frac{\mu_n}{x_n}\right) = \frac{\mu_1}{f(x_1)} + \frac{\mu_2}{f(x_2)} + \dots + \frac{\mu_n}{f(x_n)} \quad (3.16)$$

If more than one element of  $X$  is mapped by  $f$  to the same element  $y \in Y$ , then the maximum of the membership grades of these elements in the fuzzy set  $A$  is chosen as the membership grade for  $y$  in  $f(A)$ . If no element  $x \in X$  is mapped to  $y \in Y$ , then the membership grade of  $y$  in  $f(A)$  is zero. Often a function  $f$  maps ordered tuples of elements of several different sets  $X_1, X_2, \dots, X_n$  such that  $f(x_1, x_2, \dots, x_n) = y, y \in Y$ . In this case, for any arbitrary fuzzy sets on  $X_1, X_2, \dots, X_n$ , respectively, the membership grade of element  $y$  in  $f(A_1, A_2, \dots, A_n)$  is equal to the minimum of the membership grades of  $x_1, x_2, \dots, x_n$  in  $A_1, A_2, \dots, A_n$ , respectively.



In order to maintain the family of fuzzy sets, previously defined, unchanged after algebraic operations are performed, an approximation principle is used (Zadeh 1975). According to this principle if we have  $n$  fuzzy sets  $A_1, A_2, \dots, A_n \in P(X)$ , and if  $\hat{A} = f(A_1, A_2, \dots, A_n)$  and,

$$d(\mu_{\hat{A}}(x), \mu_A(x)) = \min_{B \in P(X)} d(\mu_{\hat{A}}(x), \mu_B(x)), \quad A \in P(X) \quad (3.17)$$

then the approximation of  $\hat{A}$  is  $A$ . Where  $d(., .)$  is any distance which satisfies the axioms of a metric (Shen and Leitch 1993). It is clear that the selection of a distance metric plays a critical role in the application of the approximation principle. In order to save computational time and memory storage, a metric based on appropriate features of the membership functions is usually used.

The use of the formal definition given above for the extension principle entails various types of computational difficulties (Shen and Leitch 1993). The solution to these difficulties is based on the parametric representation of the membership distribution of a fuzzy number. This parametric representation is achieved by the 4-tuple  $(a, b, \alpha, \beta)$ . The first two parameters indicate the interval in which the membership value is 1; the third and fourth parameters indicate the left and right width of the distribution. Linear functions are used to define the slopes. Therefore, the membership function  $\mu_A(x)$ , of the fuzzy number  $A = (a, b, \alpha, \beta)$  is defined as expressed in the following equation,

$$\mu_A = \begin{cases} 0 & \text{for } x < (a - \alpha) \\ \left(\frac{1}{\alpha}\right)(x - a + \alpha) & \text{for } x \in [(a - \alpha), a] \\ 1 & \text{for } x \in [a, b] \\ \left(\frac{1}{\beta}\right)(b + \beta - x) & \text{for } x \in [b, (b + \beta)] \\ 0 & \text{for } x > (b + \beta) \end{cases} \quad (3.18)$$

and Figure 3.4 shows the membership distribution of the fuzzy number quoted.

The arithmetic operations on these fuzzy numbers are well developed (Bonissone and Decker 1986), and for the preceding reasons we adopt such a representation to form the fuzzy quantity spaces used in the following chapters.

This solution is a very good approximation of the result obtained from using the extension principle to evaluate arithmetic functions with fuzzy numbers, and has a much more limited computational overhead. So, the qualitative simulation presented in further chapters is performed through the following two formulas,

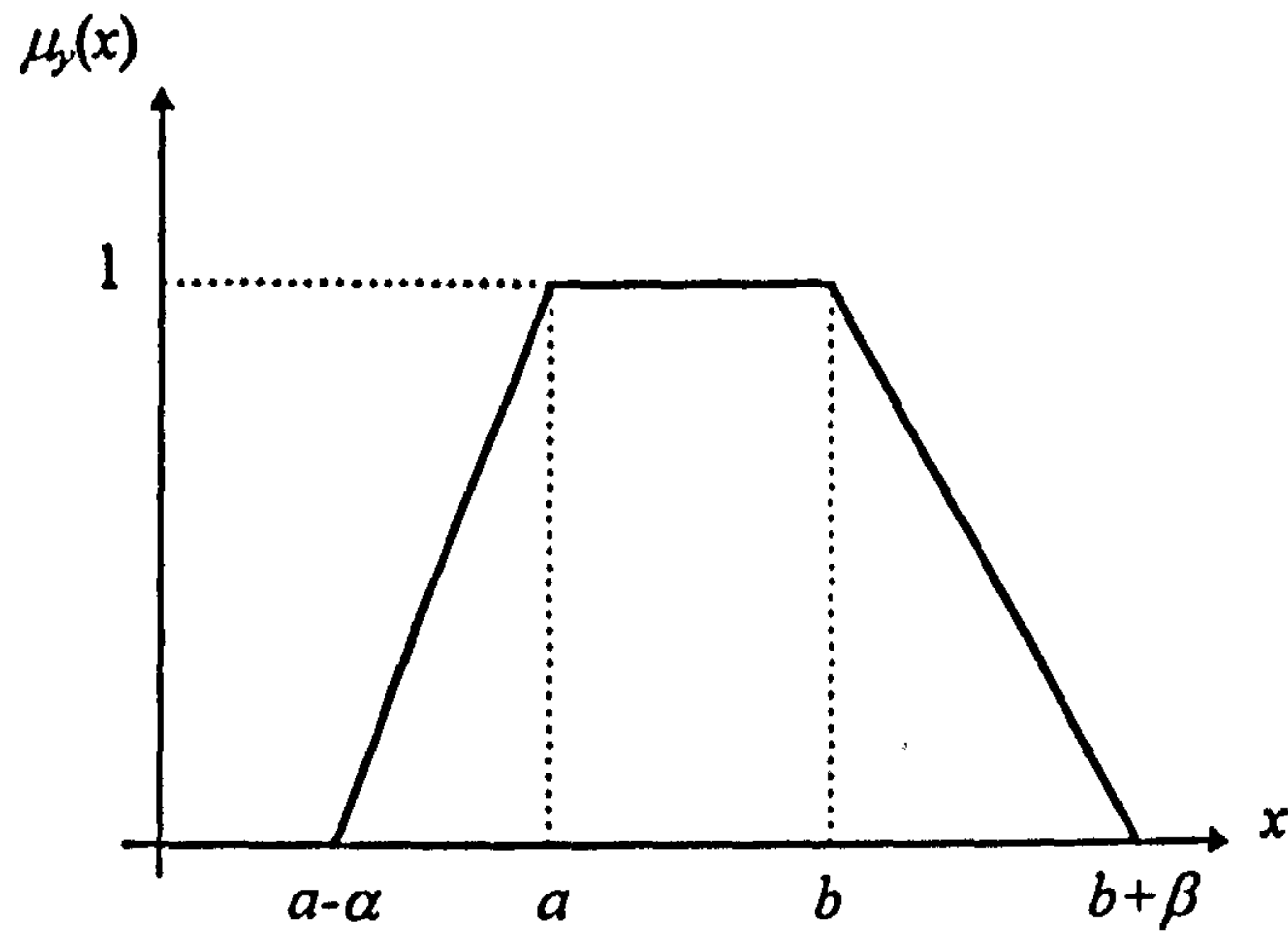


Figure 3.4 - Parametric representation of a normal convex fuzzy number

$$A + B = [a + c, b + d, \alpha + \gamma, \beta + \delta] \quad (3.19)$$

$$A - B = [a - d, b - c, \alpha + \delta, \beta + \gamma] \quad (3.20)$$

where  $A \equiv [a, b, \alpha, \beta]$  and  $B \equiv [c, d, \gamma, \delta]$ .

Two important features of a fuzzy set that are used in fuzzy mathematics are its power and centre. The power of a fuzzy set, with a normal and convex membership distribution, is defined as the integral of its membership distribution,

$$Power(A) = \int_R \mu_A(x) dx \quad (3.21)$$

The centre of  $A$  is the central element among those elements whose degrees of membership are equal to the maximum membership value.

If the membership distribution of a fuzzy number  $A$  has a parametric representation by the 4-tuple  $(a, b, \alpha, \beta)$ , the Power and Centre can be evaluated through the following formulae, respectively,

$$Power(A) = \frac{1}{2} [2(b - a) + \alpha + \beta] \quad (3.22)$$

$$Centre(A) = \frac{1}{2} (a + b) \quad (3.23)$$



Moreover, the fuzzy qualitative simulation algorithm described in a further chapter uses the approximation principle previously introduced to evaluate the degree of closeness between fuzzy sets. Therefore, in order to calculate the degree of closeness between two fuzzy sets, each of which belongs to a different subset of the same universe, the distance measure defined by the following equation (Shen and Leitch 1993), is used,

$$d(A, B) = \sqrt{\left( \underset{A \in q_f}{Power(A)} - \underset{B \in \hat{q}_f}{Power(B)} \right)^2 + \left( Centre(A) - Centre(B) \right)^2} \quad (3.24)$$

In the last equation, the quantity space  $q_f$  is one of the two subsets, while another one, denoted  $\hat{q}_f$ , is the collection of all results from operations applied to the elements among  $q_f$ . Once we have a desirable distance measure, the approximation of the fuzzy number  $B \in \hat{q}_f$ , to a qualitative value  $A$  in  $q_f$  can be determined by choosing  $A$  such that the distance between  $B$  and  $A$  is the smallest among all the distances between the fuzzy number  $B$  and all elements in  $q_f$ . In the case when are more than one value in  $q_f$  which have the same shortest distance from  $B$ , all such values are treated as the approximation results of the original calculation.

## **Chapter 4**

# **Modelling and Rule Based Control of a Mixing Process**

### **4.1 Introduction**

In order to investigate the application of different artificial intelligence techniques to on-line process control, as well as to on-line fault detection and diagnosis, a mixing process has been taken as an example of an industrial process. Several real time expert systems, including a rule based controller and various different on-line fault detection and diagnosis systems, have been developed for this process. Therefore, at an initial stage of this research, a dynamic mathematical model for the mixing process has been derived, which is presented in sub chapter 4.3. From this mathematical model a qualitative one was developed for the mixing process, which will be used in following chapters for fault detection purposes. This qualitative model is constituted by a set of confluence's which are presented in sub chapter 4.4.

The first expert system developed during this research is a rule based on-line control system for the mixing process. This controller has been implemented for a multi-input and multi-output situation. It derives control actions from the causal relations among process variables, where the causal relations form a symbolic model of the process. Since the symbolic model captures the causal relations inside the system, for some situations, it can be more understandable than any numerical model. The rule based control system is described in sub chapter 4.5, where the causal relations in the mixing process and the control rules are described in detail. The performance of the rule based controller is discussed in sub chapter 4.6, where some results achieved during simulation studies are also presented. The last sub chapter contains some concluding remarks.



# 4.2 The Mixing Process

The layout of the mixing process is presented in Figure 4.1. Two tanks in cascade receive hot and cold water input streams. The hot water, at about 80 °C, is supplied from an electrically heated tank, while the cold water is supplied from the mains. Both streams enter tank 1 where mixing takes place. The contents of tank 1 pass to tank 2 and subsequently out to the pool tank from which they are recycled to the header tank. A number of hand valves can be seen in the mixing process shown in Figure 4.1. These hand valves are either kept fully open or fully closed during normal operation, as their function is simply to allow different systems configuration. When both tanks are used, as it has been considered during this research, hand valves 1, 2, 3 and 5 are fully open and hand valve 4 is closed. The mixing process, although simple in operation, enables generic concepts to be developed. The simplicity of the process does not obscure the fundamental basic ideas which are being studied.

Measurement of the process variables level and temperature in both tanks is available and, hence, it is possible to control level and temperature in either tank. However, during the research work conducted with this process only the case of tank 2 being controlled has been considered.

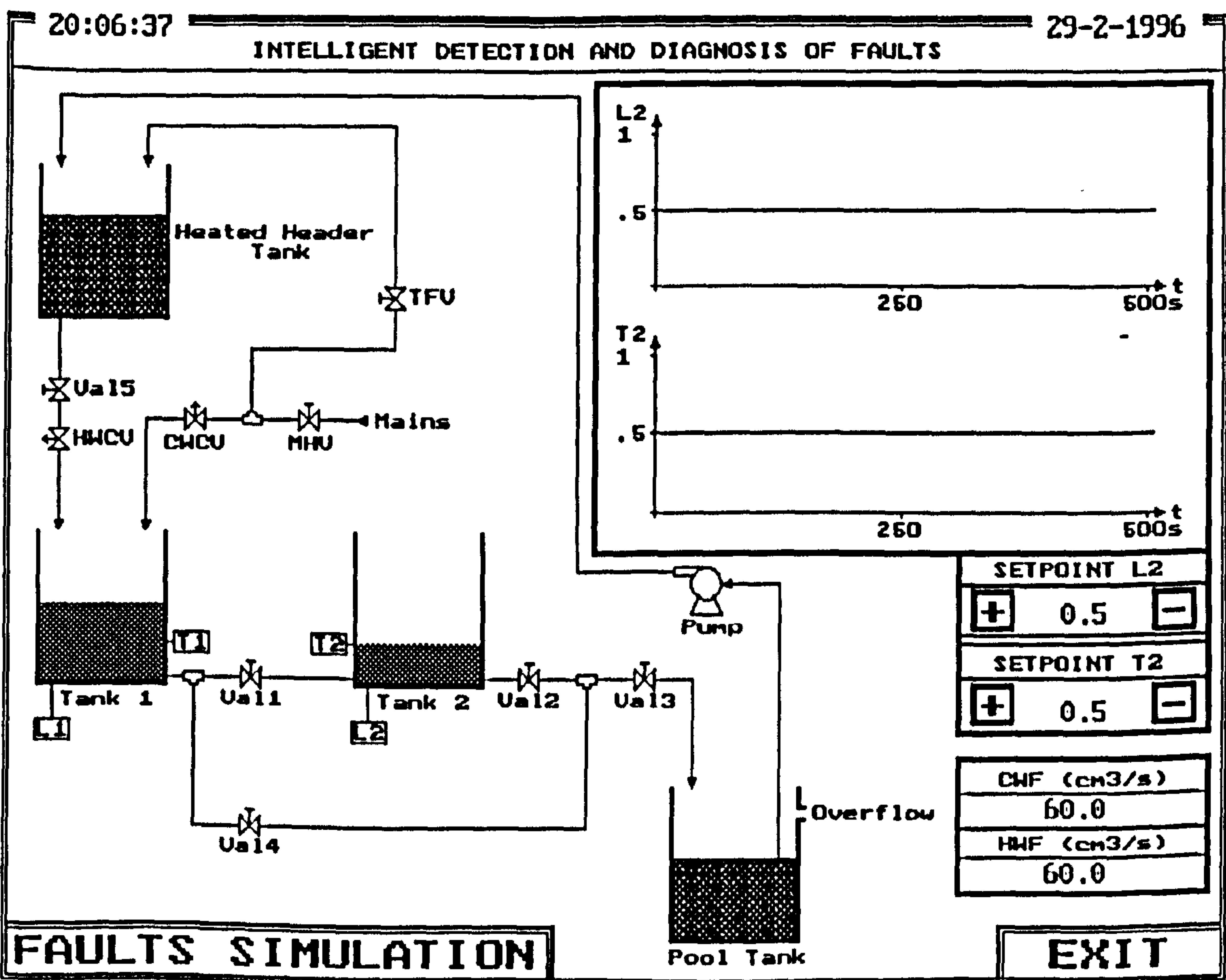


Figure 4.1 - The mixing process

The dynamic model for the mixing process is developed in the next sub chapter, which is achieved from mass and heat balances relationships performed in tanks 1 and 2 respectively.

### 4.3 The Mixing Process Dynamic Model

This sub chapter presents the dynamic model for the mixing process. This mathematical model will be used to simulate the process under normal operation conditions, as well as under failure situations. As quoted above the model is determined from mass and heat balance relationships. Thus, from the mass balance in tank 1 of the mixing process, the following equation can be obtained,

$$\frac{d(A_1 L_1 \rho)}{dt} = \rho(Q_c + Q_h) - \rho Q_{o_1} \quad (4.1)$$

Which can be simplified to,

$$A_1 \frac{dL_1}{dt} = Q_c + Q_h - Q_{o_1} \quad (4.2)$$

From the application of the mass balance to tank 2, the following equation, (4.3), can be obtained,

$$\frac{d(A_2 L_2 \rho)}{dt} = \rho(Q_{o_1} - Q_{o_2}) \quad (4.3)$$

Which can be simplified to,

$$A_2 \frac{dL_2}{dt} = Q_{o_1} - Q_{o_2} \quad (4.4)$$

The heat balance in tank 1 can be represented as follows,

$$\frac{d(C_p A_1 L_1 T_1)}{dt} = C_p Q_c T_c + C_p Q_h T_h - C_p Q_{o_1} T_1 \quad (4.5)$$



The equation (4.5) can be simplified such that the following equation can be obtained,

$$A_1 T_1 \frac{dL_1}{dt} + A_1 L_1 \frac{dT_1}{dt} = Q_c T_c + Q_h T_h - Q_{o_1} T_1 \quad (4.6)$$

Multiplying the two sides of equation (4.2) by  $T_1$ , and then substituting the result into equation (4.6) gives,

$$A_1 L_1 \frac{dT_1}{dt} = Q_c (T_c - T_1) + Q_h (T_h - T_1) \quad (4.7)$$

From the heat balance in tank 2 of the mixing process, the following equation, (4.8), has been obtained,

$$\frac{d(C\rho L_2 T_2)}{dt} = C\rho Q_{o_1} T_1 - C\rho Q_{o_2} T_2 \quad (4.8)$$

Which can be simplified to,

$$A_2 T_2 \frac{dL_2}{dt} + A_2 L_2 \frac{dT_2}{dt} = Q_{o_1} T_1 - Q_{o_2} T_2 \quad (4.9)$$

Multiplying both sides of equation (4.4) by  $T_2$ , and then substituting it into equation (4.9), gives,

$$A_2 L_2 \frac{dT_2}{dt} = Q_{o_1} (T_1 - T_2) \quad (4.10)$$

The output flows from the two tanks,  $Q_{o_1}$  and  $Q_{o_2}$ , are determined by pressure differences and valve parameters, and can be expressed as follows,

$$Q_{o_1} = k_1 \sqrt{L_1 - L_2} \quad (4.11)$$

$$Q_{o_2} = k_2 \sqrt{L_2} \quad (4.12)$$

In short briefly, the dynamic model of the mixing process is presented below by equations (4.13) to (4.18),

$$A_1 \frac{dL_1}{dt} = Q_c + Q_h - Q_{o1} \quad (4.13)$$

$$A_2 \frac{dL_2}{dt} = Q_{o1} - Q_{o2} \quad (4.14)$$

$$A_1 H_1 \frac{dT_1}{dt} = Q_c (T_c - T_1) + Q_h (T_h - T_1) \quad (4.15)$$

$$A_2 H_2 \frac{dT_2}{dt} = Q_{o1} (T_1 - T_2) \quad (4.16)$$

$$Q_{o1} = k_2 \sqrt{L_1 - L_2} \quad (4.17)$$

$$Q_{o2} = k_2 \sqrt{L_2} \quad (4.18)$$

where the following notation is used,

$A_1$  - is the cross-sectional area of tank 1 ( $cm^2$ );

$L_1$  - is the level in tank 1 ( $cm$ );

$T_1$  - is the temperature of water in tank 1 ( $^{\circ}C$ );

$A_2$  - is the cross-sectional area of tank 2 ( $cm^2$ );

$L_2$  - is the level in tank 2 ( $cm$ );

$T_2$  - is the temperature of water in tank 2 ( $^{\circ}C$ );

$T_c$  - is the temperature of input cold water ( $^{\circ}C$ );

$T_h$  - is the temperature of input hot water ( $^{\circ}C$ );

$Q_c$  - is the input cold water flow rate ( $cm^3/sec$ );

$Q_h$  - is the input hot water flow rate ( $cm^3/sec$ );

$Q_{o1}$  - is the output flow rate from tank 1 to tank 2 ( $cm^3/sec$ );

$Q_{o2}$  - is the output flow rate from tank 2 ( $cm^3/sec$ );

$t$  - is the time ( $sec$ );



$\rho$  - is the density of the inlet water ( $g/cm^3$ );

$C$  - is the specific heat of the inlet water ( $J/g^{\circ}C$ ).

Moreover, during the simulation studies conducted with the dynamic model just derived, the values used for the independently variables have been,

$$A_1 = 285.6 \text{ cm}^2;$$

$$A_2 = 150.04 \text{ cm}^2;$$

$$T_h = 80 \text{ }^{\circ}C;$$

$$T_c = 20 \text{ }^{\circ}C.$$

The other two unknown parameters,  $k_1$  and  $k_2$ , are determined from experiments, through the least squares estimation algorithm as presented by Zhang (1991). He obtained the following values for these parameters:

$$k_1 = 29.07 \text{ cm}^{5/2}/\text{sec};$$

$$k_2 = 29.46 \text{ cm}^{5/2}/\text{sec}.$$

In the mixing process dynamic model just achieved, the following assumptions have been made:

- Difference between cold and hot water density is negligible;
- Difference between cold and hot water specific heat is negligible;
- The mixing is perfect;
- Water doesn't boil;
- Heat transfer coefficients are constant;
- Heat losses into the environment are negligible.

The qualitative model presented in the next sub chapter is derived from the quantitative one presented here.

## 4.4 Qualitative Modelling of the Mixing Process

The strategy for developing a qualitative representation is to search for a qualitative mathematics capable of yielding significant results from a minimum of information. This sub chapter presents the qualitative model, which has been derived from the quantitative model presented in the last sub chapter, for the process under consideration.

As quoted above, several different approaches to qualitative modelling are pointed out by several researchers. The qualitative model presented by the author is a set of confluence's, which consists of a set of qualitative equations derived from the quantitative model under concern, such as in de Kleer and Brown's confluence based qualitative reasoning approach (de Kleer and Brown 1984). Therefore, the qualitative model or confluence's of the mixing process, shown below, can be obtained through the dynamic model linearized by comparing the dynamic model at a present state with that at a previous state, or by Taylor series expansion.

However, the sign of some variables are dependent on the process state. In order to solve this problem, we have used the approach presented by Shiozaki *et al.* (1985) and Kramer and Palowitch (1987). This means, that a conditional sign,  $x$ , of a variable,  $A$ , can be expressed as follows:

$$\begin{aligned} x &= + && \text{if } cond_1; \\ x &= - && \text{if } cond_2; \\ x &= 0 && \text{otherwise.} \end{aligned}$$

Thus, the qualitative model of the mixing process, can be expressed as,

$$[\Delta \dot{L}_1] = [\Delta Q_c] + [\Delta Q_h] - [\Delta Q_{o1}] \quad (4.19)$$

$$[\Delta \dot{L}_2] = [\Delta Q_{o1}] - [\Delta Q_{o2}] \quad (4.20)$$

$$[\Delta \dot{T}_1] = [\Delta Q_h] - [\Delta Q_c] - [\Delta T_1] \quad (4.21)$$

$$\begin{aligned} &\text{if } T_1 > T_2 \\ [\Delta \dot{T}_2] &= [\Delta Q_{o1}] + [\Delta T_1] - [\Delta T_2] \end{aligned} \quad (4.22)$$



$$\begin{aligned} \text{if } T_1 < T_2 \\ [\Delta \dot{T}_2] &= -[\Delta Q_{o1}] + [\Delta T_1] - [\Delta T_2] \end{aligned} \quad (4.23)$$

$$\begin{aligned} \text{if } T_1 = T_2 \\ [\Delta \dot{T}_2] &= [\Delta T_1] - [\Delta T_2] \end{aligned} \quad (4.24)$$

where the qualitative values of  $[\Delta Q_{o1}]$  and  $[\Delta Q_{o2}]$ , are evaluated through the following two equations,

$$[\Delta Q_{o1}] = [\Delta(L_1 - L_2)] \quad (4.25)$$

$$[\Delta Q_{o2}] = [\Delta L_2] \quad (4.26)$$

## 4.5 Rule Based Control of the Mixing Process

There are many industrial examples where the high-level of supervisory, control and optimisation of manufacturing processes is not automated but is conducted by a combination of process experts and plant operators. In some cases, this may be due to safety or other operational constraints deliberately imposed by management. Often, however, this level of the process has not been automated due to complexity. The types of problems that can exist are that many of the key control parameters are not capable of being measured directly or reliably, and many processes are inherently multivariable in nature. This means that changes in one manipulated variable can cause changes to the whole processes, and often the response to such changes is highly non-linear. When this is coupled with the possibility of random disturbances, and long transport lags that can mask the effects of control changes, the problem can become extremely complex.

Conventional mathematical modelling has succeeded best in the automation of those processes which are well understood and well behaved. However, in a number of more complex applications it has failed to be sufficiently robust to provide a reliably high level of automation. In these situations a rule base controller may be more adequate. The paradox is that in many of the processes where mathematical modelling has not proven effective, the process is controlled by one or two human operators, often apparently with a great deal of ease. Closer examination of the techniques used by such skilled operators to control such a complex process shows that they rely on some basic process knowledge (but little deep knowledge) combined with heuristic rules acquired

through experience. Attempting to mimic the human expert therefore provides a natural and interesting alternative to mathematical modelling of the process; in some cases, it may be the only alternative.

The reasons quoted above motivated the present study, where the controller is designed based on causal relationships between subsystems, and the control actions are inferred from samples of both controlled and non-controlled variables. In the mixing process, the level and temperature of tank 2 are directly affected by those of tank 1, and the level and temperature of tank 1 are directly affected by the inlet hot and cold flow streams. These causal relationships are used to infer control actions.

Based on steady-state conditions, an increase in inlet flow will cause the level in the tank 1 to increase, whereas a decrease in inlet flow will cause the level in tank 1 to decrease. An increase in inlet hot flow or a decrease in inlet cold flow will cause the temperature of tank 1 to increase. An increase in level and temperature of tank 1 will cause the level and temperature of tank 2 to increase respectively, and a decrease in level and temperature of tank 1 will cause the level and temperature of tank 2 to decrease respectively. These causal relationships form a symbolic model of the system.

Based on the symbolic model and the current state of the system, control actions can be inferred. The control rules are in the following form,

$$Goal \wedge Condition \Rightarrow Subgoal$$

where,

- *Goal*, is the destination to be achieved;
- *Condition*, is the current state;
- *Subgoal*, is the intermediate goal to be achieved under the particular "*Condition*" in order to achieve the "*Goal*".

The sets of rules presented below are similar to those pointed out by Zhang (1991), but here a different approach is used. While in Zhang's approach a linguistic description of the behaviour of the variables is used, the author performs the rule based controller from the proposal of de Kleer and Brown (de Kleer and Brown 1984), where the behaviour of the process variables is represented through its qualitative values.

Therefore, in the approach presented by the author the aim is to reduce the quantitative precision of the behaviour description but retain the important distinctions. Instead of continuous



real-valued variables, each variable is described qualitatively. Since one of the most important features of a physical variable is whether it is increasing, unchanging or decreasing; +, 0 and – are defined as the quantity space, where +, 0 and – represent the cases that a variable is increasing, unchanging and decreasing respectively. For instance, the following rule:

$$-[L_2] = 0 \wedge [L_2] = + \Rightarrow [L_1] = -;$$

can be interpreted as "To achieve a qualitative value of the level in the tank 2 unchanging while the qualitative value of the level in the tank 2 is increasing, the qualitative value of the level in the tank 1 should be decrease".

Since for the level and temperature control loops the symbolic models are identical, they have the same control rules. The full sets of rules are listed below,

#### Rule set 1:

$$Y_2 = \text{Setpoint} \wedge Y_2 < \text{Setpoint} \Rightarrow [Y_2] = +$$

$$Y_2 = \text{Setpoint} \wedge Y_2 = \text{Setpoint} \Rightarrow [Y_2] = 0$$

$$Y_2 = \text{Setpoint} \wedge Y_2 > \text{Setpoint} \Rightarrow [Y_2] = -$$

#### Rule set 2:

$$[Y_2] = + \wedge [Y_2] = - \Rightarrow [Y_1] = +$$

$$[Y_2] = + \wedge [Y_2] = 0 \Rightarrow [Y_1] = +$$

$$[Y_2] = + \wedge \{[Y_2] = +, Y_1 < A\} \Rightarrow [Y_1] = +$$

$$[Y_2] = + \wedge \{[Y_2] = +, Y_1 > A\} \Rightarrow [Y_1] = 0$$

The parameter A take a value slightly lower than the steady state value of  $Y_1$  corresponding to the setpoint of  $Y_2$ .

#### Rule set 3:

$$[Y_2] = 0 \wedge [Y_2] = + \Rightarrow [Y_1] = -$$

$$[Y_2] = 0 \wedge [Y_2] = 0 \Rightarrow [Y_1] = 0$$

$$[Y_2] = 0 \wedge [Y_2] = - \Rightarrow [Y_1] = +$$

**Rule set 4:**

$$[Y_2] = - \wedge [Y_2] = + \Rightarrow [Y_1] = -$$

$$[Y_2] = - \wedge [Y_2] = 0 \Rightarrow [Y_1] = -$$

$$[Y_2] = - \wedge ([Y_2] = -, Y_1 > B) \Rightarrow [Y_1] = -$$

$$[Y_2] = - \wedge ([Y_2] = -, Y_1 < B) \Rightarrow [Y_1] = 0$$

The parameter  $B$  take a value slightly higher than the steady state value of  $Y_1$  corresponding to the setpoint of  $Y_2$ .

**Rule set 5:**

$$[Y_1] = + \wedge [Y_1] = - \Rightarrow [Q] = +$$

$$[Y_1] = + \wedge [Y_1] = 0 \Rightarrow [Q] = +$$

$$[Y_1] = + \wedge [Y_1] = + \Rightarrow [Q] = 0$$

**Rule set 6:**

$$[Y_1] = 0 \wedge [Y_1] = + \Rightarrow [Q] = -$$

$$[Y_1] = 0 \wedge [Y_1] = 0 \Rightarrow [Q] = 0$$

$$[Y_1] = 0 \wedge [Y_1] = - \Rightarrow [Q] = +$$

**Rule set 7:**

$$[Y_1] = - \wedge [Y_1] = + \Rightarrow [Q] = -$$

$$[Y_1] = - \wedge [Y_1] = 0 \Rightarrow [Q] = -$$

$$[Y_1] = - \wedge [Y_1] = - \Rightarrow [Q] = 0$$

When dealing with the level control loop,  $Y_1$ ,  $Y_2$  and  $Q$  stand for level in tank 1, level in tank 2 and inlet cold flow respectively. When dealing with the temperature control loop,  $Y_1$ ,  $Y_2$  and  $Q$  stand for temperature in tank 1, temperature in tank 2 and inlet hot flow respectively. Within the rule sets, the changes in inlet cold flow and/or inlet hot flow are proportional to the error between the desired value and sampled value, with a proportional parameter  $K$ . In order to obtain a quick response the parameters  $A$  and  $B$  are introduced in rule sets 2 and 4 (Zhang 1991).



According to the previous work on controlling this mixing process (Ellis *et al.* 1986, Zhang 1991), the hot inlet flow is used to control temperature while the cold inlet flow is used to control level. Since either hot inlet flow or cold inlet flow can affect both temperature and level, interaction exists between the two control loops. It is necessary to design a decoupling scheme to eliminate the interaction.

Hence, after the control actions for the individual loops have been inferred from the above control rules, they should be modified in order to eliminate the interactions. To do this, two situations must be considered. The first is when the hot water flow is changing while the cold water flow is kept steady. Here, in order to eliminate the effect of changing hot water flow on the level control loop, the total amount of inlet water flow should be unchanged. That is,

$$\Delta Q_c + \Delta Q_h = 0 \quad (4.27)$$

Therefore,

$$\Delta Q_c = -\Delta Q_h \quad (4.28)$$

Then, in the situation that the hot water flow is changing while the cold flow is kept steady, the final control action is that the cold water inlet flow should be changed by the following quantity,  $-\Delta Q_h$ .

The other situation is when the hot water inlet flow is kept unchanged while the cold water inlet flow is changing. Here, in order to eliminate the effect of changing cold water inlet flow on the temperature control loop, the total input heat should be unchanged. This can be expressed by the following equation,

$$\Delta Q_c C(T_c - T_1)\rho + \Delta Q_h C(T_h - T_1)\rho = 0 \quad (4.29)$$

Therefore,

$$\Delta Q_h = \frac{(T_1 - T_c)\Delta Q_c}{T_h - T_1} \quad (4.30)$$

Thus, when the hot water inlet flow is kept unchanged while the cold water inlet flow is changing, the final control action is that the hot water inlet flow should be changed by  $(T_1 - T_c)\Delta Q_c / (T_h - T_1)$ .

In the implementation of the rule base controller, the knowledge quoted above, is represented by rules in the form "IF *antecedent* THEN *consequence*", such that, if the situation described in the antecedent part of the rule is true, then the action specified in the consequence part is produced. To fire the rules, a inference engine with forward chaining abilities has been implemented. This starts with a set of facts or given data and then searches the knowledge base for rules whose "IF" portion match the data. This generates new facts and data in the knowledge base which in turn causes other rules to fire. The reasoning operation stops when no more rules can be fired. This kind of reasoning is also known as data-driven inference (Jackson 1986).

Sub chapter 4.6 presents some results achieved during simulation studies of the implemented rule based controller, where it can be seen that a quite successful performance has been obtained.

### 4.6 Performance of the Rule Based Controller

The rule based controller has been implemented using a TURBO C++ program running in any PC without special features. Its performance is very satisfactory, as can be seen from the Figures 4.2 and 4.3, for steps changes in the setpoints of the controlled variables temperature and

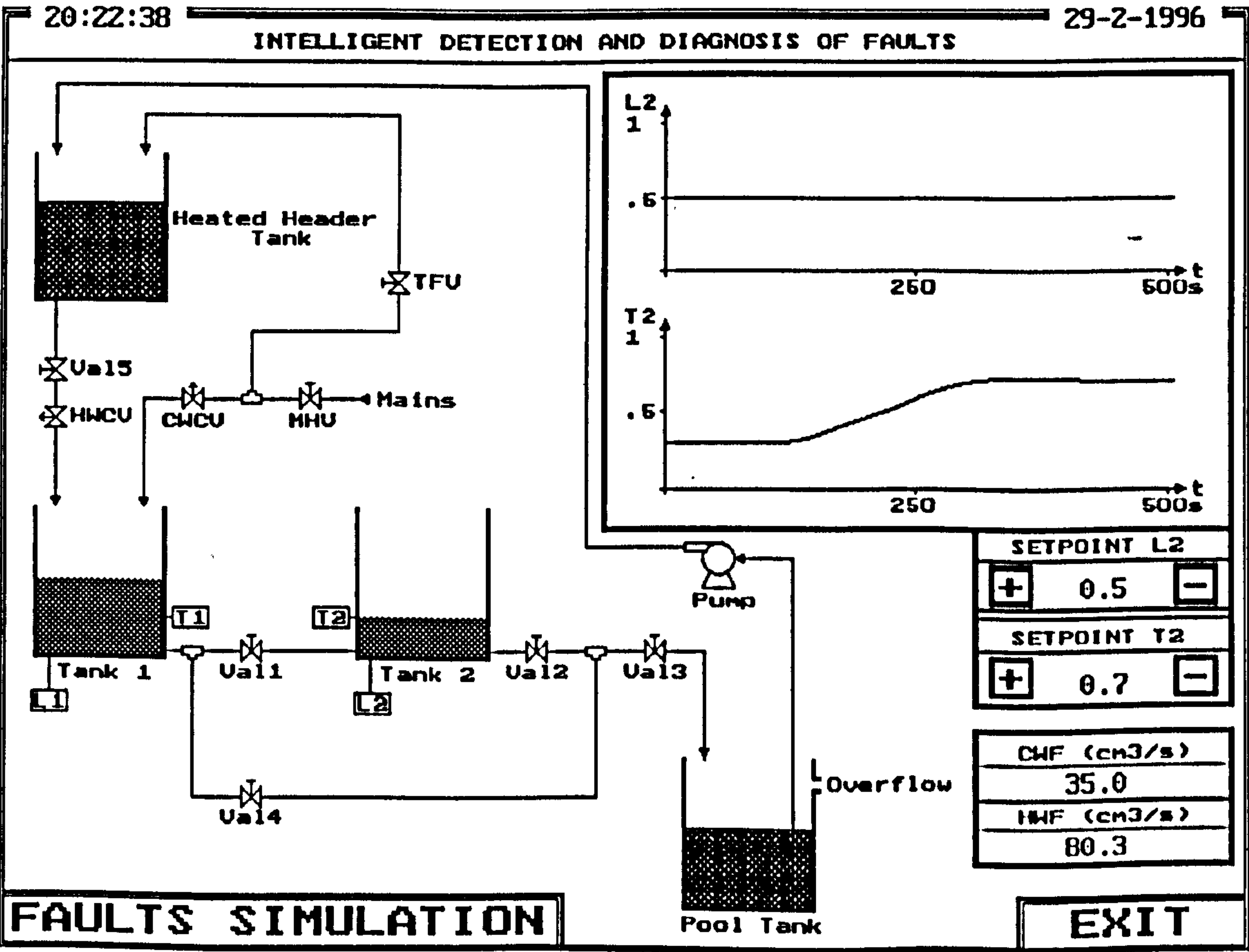


Figure 4.2 - Controller response when a step in the setpoint of temperature is performed



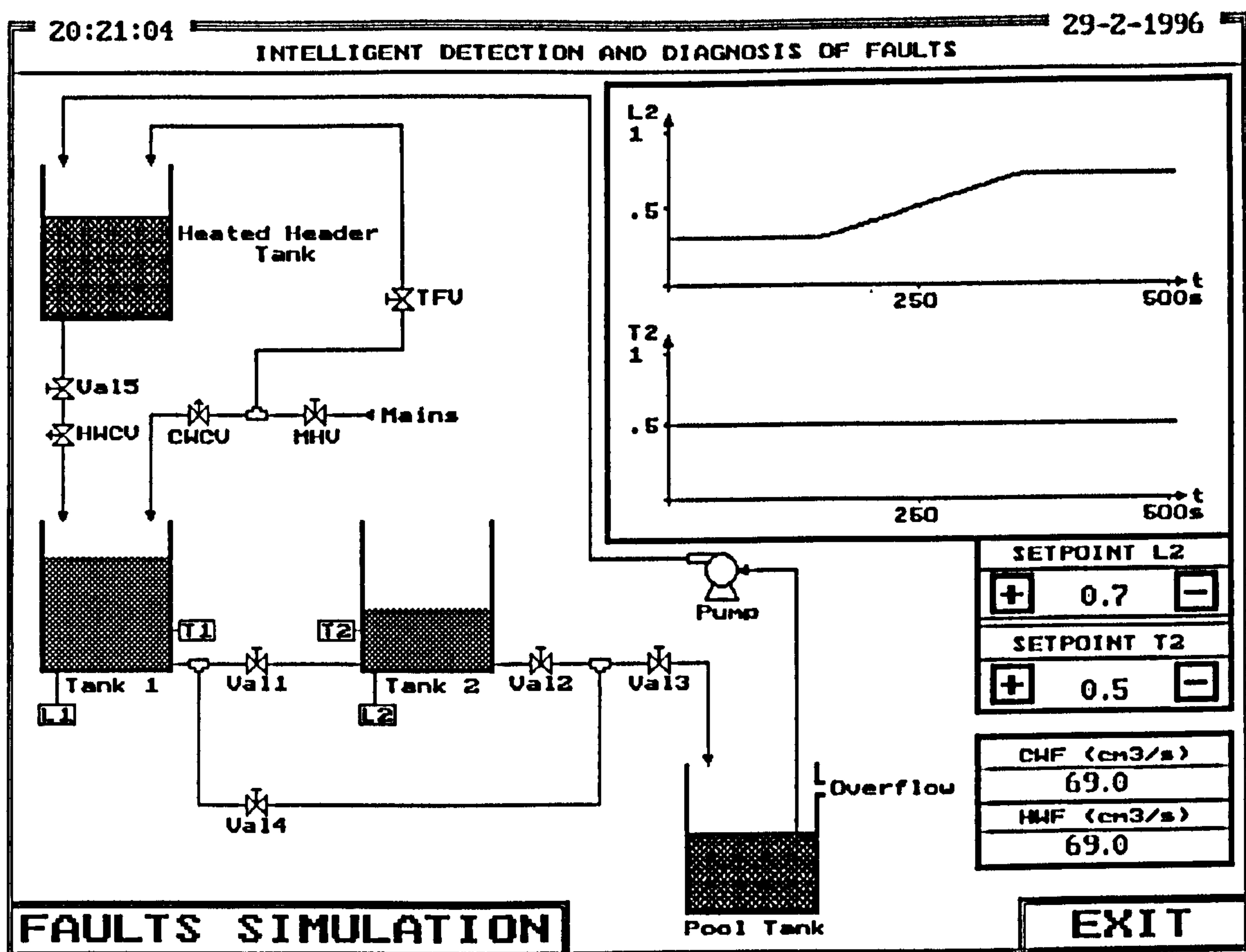


Figure 4.3 - Controller response when a step in the setpoint of level is performed

level in tank 2, respectively. The response of the rule base controller has low overshoot and undershoot and no interaction is observed between the two control loops. Also it is observed that the rule base controller has settled to the steady state condition in about 200 seconds real time.

The tuning of the rule base controller is performed by a trial-error procedure by adjusting the parameters  $K$ ,  $A$  and  $B$ , and is relatively easy. It has been found that the controller is not very sensitive to changing the tuning parameters. This suggests that the properties of a rule based controller is largely determined by its rules. The role of controller parameters is less crucial in rule based controllers than in conventional controllers. Therefore, one can conclude that a rule base controller could be an alternative for conventional controllers in cases where numerical models for the controlled processes are not available or are difficult to obtain, or in cases where the key control parameters are not capable of being measured directly or reliably.

## 4.7 Conclusions

From mass and heat balance relationships the dynamic model for the mixing process is derived in sub chapter 4.3. From this dynamic mathematical model a qualitative model has been

achieved which is presented in sub chapter 4.4. Both models will be used in subsequent chapters, as a test bed of some artificial intelligence techniques proposed for industrial process fault detection and diagnosis.

In sub chapter 4.5 a rule based controller is presented, whose performance has been discussed in sub chapter 4.6. Control of level and temperature in tank 2 has been performed through the rule based controller and a satisfactory performance has been observed. Moreover, it has been observed that the properties of a rule based controller are mainly determined by its rules, and that a rule based controller is not very sensitive to the changes in its parameters. This demonstrates the robustness of rule based controllers and, hence, suggests that they can be an alternative for conventional controllers in cases where conventional mathematical modelling fails to provide a high level of automation.



## **Chapter 5**

# **Modelling and Control of a Continuous Stirred Tank Reactor**

### **5.1 Introduction**

In an operating chemical plant, product quality is maintained by monitoring process variables and controlling their fluctuations within a desired range. When operating conditions vary outside these design limits, not only is product quality in danger but, if left uncorrected these variations could result in a catastrophic event such as an explosion, fire, or the release of toxic chemicals. The operator's task in the event of a process malfunction is to diagnose the cause of the plant upset quickly and accurately so that corrective action may be taken in time. However, diagnosis of process malfunctions is a difficult task for process operators. Under certain circumstances, operators may have difficulty handling unanticipated events or low probability failures. In this context, there was motivation to choose a chemical process as another example of an industrial plant, which has also been used as test bed of various on-line fault detection and diagnosis systems presented in following chapters.

Therefore, a continuous stirred tank reactor (CSTR) has been selected, similar to that used by Kramer and co-authors (Kramer and Palowitch 1987, Finch and Kramer 1988, Oyeleye and Kramer 1988, Kramer and Finch 1989), as well as by Zhang and co-authors (Zhang 1991, Zhang and Roberts 1991a, Zhang and Roberts 1992a, Zhang and Roberts 1992b, Zhang and Morris 1994). Such a process is described in sub chapter 5.2. In sub chapter 5.3, a dynamic model for the process is derived. From this mathematical model a qualitative model is developed for the CSTR process, whose details are described in sub chapter 5.4. During simulation studies of the CSTR

plant control of some process variables has been considered. Thus, in sub chapter 5.5 some features of the controller used to perform the regulation tasks are presented. Some concluding remarks about the performance of the controllers implemented are presented in sub chapter 5.6.

## 5.2 Continuous Stirred Tank Reactor

The continuous stirred tank reactor is a very common process in the chemical industry although it does have numerous variations regarding the introduction and extraction of energy and materials (Franks 1972). The layout of the CSTR plant used during this research, is shown in Figure 5.1, where it is assumed that the reaction takes place in the reactor vessel in isothermal conditions. Moreover, the reaction is cooled by recycle through an external heat exchanger. Temperature and level in the reactor, as well as the recycle flow rate, are controlled by feedback control systems (cascade control for the case of temperature). Classical controllers have been used to perform the process variables control. The performance of such a controllers is discussed in further sub chapters.

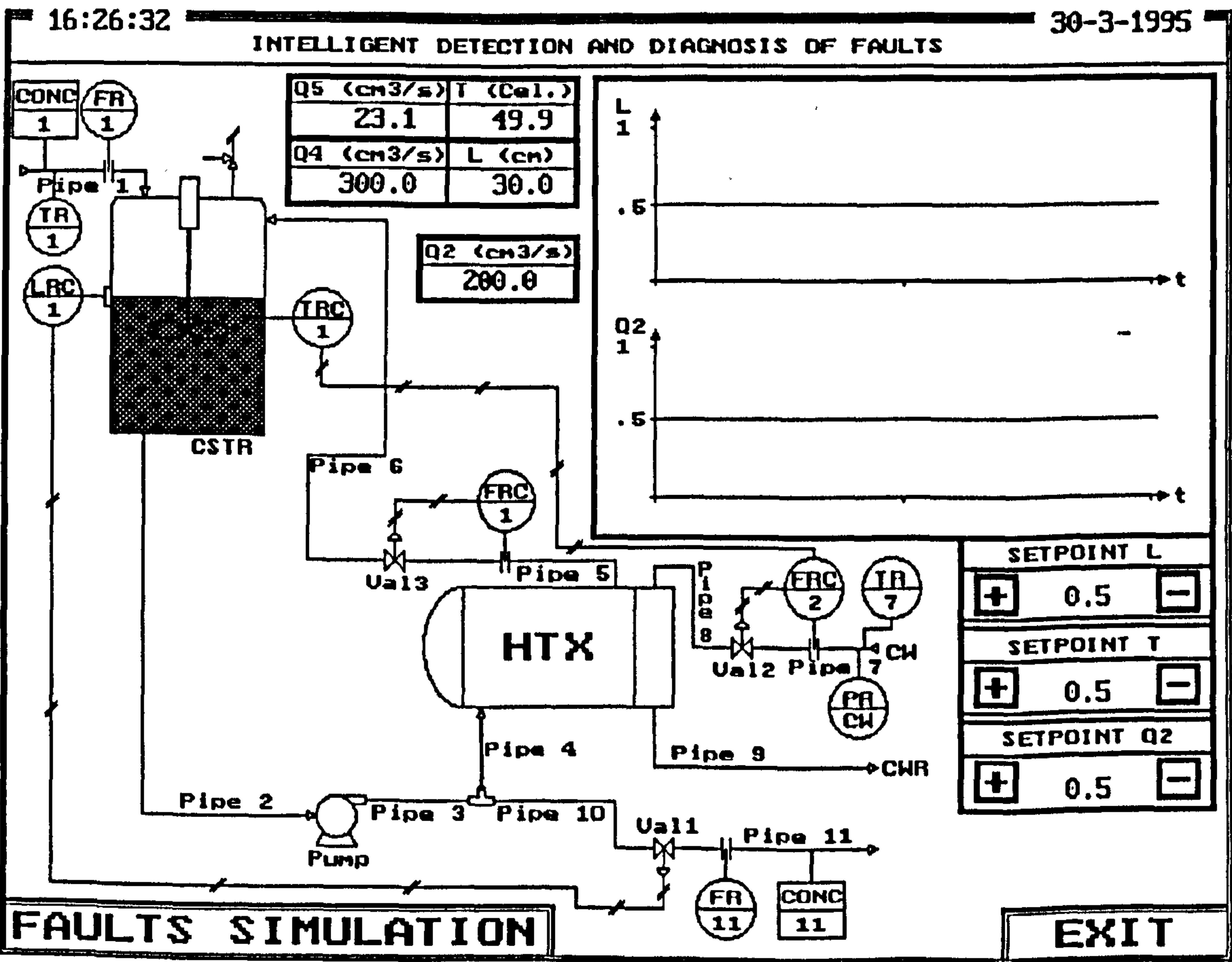


Figure 5.1 - Continuous stirred tank reactor



## 5.3 The Continuous Stirred Tank Reactor Dynamic Model

The dynamic model for the continuous stirred tank reactor is developed in this sub chapter, which is achieved based upon results presented in Franks (1972). The model is used to simulate the process under normal operating conditions and serves to obtain the behaviour of the process under a fault or set of faults situation.

Figure 5.2 shows an elementary diagram of the continuous stirred tank reactor with an input fluid feed, whose rate is  $Q_i$  ( $\text{cm}^3/\text{s}$ ) and an output flow having a rate  $Q_o$  ( $\text{cm}^3/\text{s}$ ). Several assumptions have been made in modelling the system. The underlying assumption for a continuous stirred tank reactor is that the outlet flow has the same temperature and composition as the reactor contents.

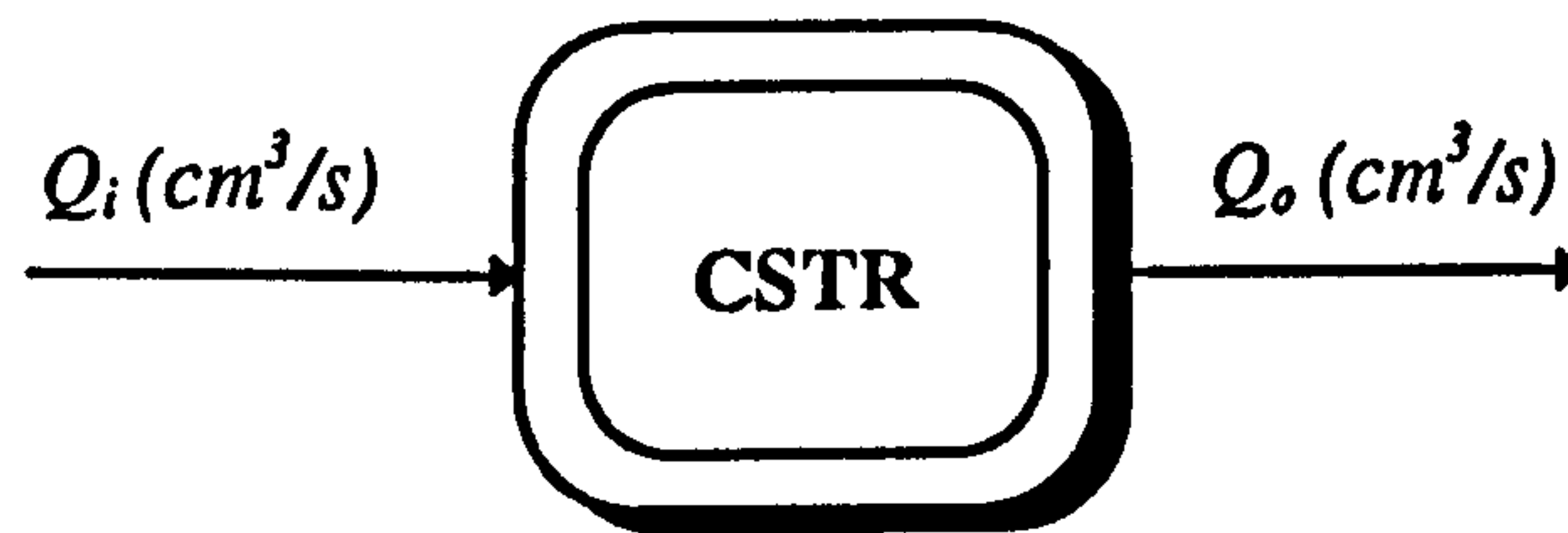


Figure 5.2 - Elementary diagram of the CSTR system

It is also assumed that perfect mixing takes place in the reactor vessel, perfect heat exchange takes place in the heat exchanger and heat losses into the environment are negligible, as well as the reactant and the product have the same density and specific heat. The model is achieved from mass and heat balances relationships in the process under concern and is constituted by the following set of differential and algebraic equations,

$$A \frac{dL}{dt} = Q_1 + Q_2 - Q_3 \quad (5.1)$$

$$AL \frac{dC_a}{dt} = Q_1(C_{a0} - C_a) - r_a AL \quad (5.2)$$

$$AL \frac{dC_b}{dt} = r_a AL - C_b Q_1 \quad (5.3)$$

$$ALB_2 \frac{dT}{dt} = B_1 Q_1(T_1 - T) - B_2 Q_2(T - T_2) + H_r r_a \quad (5.4)$$

$$B_1 = C_{a0}\rho C + (1 - C_{a0})\rho_0 C_0 \quad (5.5)$$

$$B_2 = \rho C(C_a + C_b) + (1 - C_a - C_b)\rho_0 C_0 \quad (5.6)$$

$$r_a = K_r C_a^n \quad (n > 0) \quad (5.7)$$

$$K_r = a_r e^{-b_r/T} \quad (5.8)$$

$$Q_2 = K_2 A_2 \sqrt{P} \quad (5.9)$$

$$Q_4 = K_4 A_4 \sqrt{P} \quad (5.10)$$

$$Q_5 = K_5 A_5 \sqrt{P_5} \quad (5.11)$$

$$Q_3 = Q_2 + Q_4 \quad (5.12)$$

$$P = P_0 + \Delta P \quad (5.13)$$

$$P_0 = L[(C_a + C_b)\rho + (1 - C_a - C_b)\rho_0] \quad (5.14)$$

$$T_2 = \frac{C_0 \rho_0 Q_5 T_5 + Q_2 T [C \rho (C_a + C_b) + C_0 \rho_0 (1 - C_a - C_b)]}{C_0 \rho_0 Q_5 + Q_2 [C \rho (C_a + C_b) + C_0 \rho_0 (1 - C_a - C_b)]} \quad (5.15)$$

where the following notation is used,

$L$  - level in the reactor ( $cm$ );

$T$  - temperature in the reactor ( $^{\circ}C$ );

$T_1$  - temperature of input reactant ( $^{\circ}C$ );

$T_2$  - temperature of the recycled reactant after heat exchange ( $^{\circ}C$ );

$T_5$  - temperature of cold water entering heat exchanger ( $^{\circ}C$ );

$A$  - cross-sectional area of the reactor ( $cm^2$ );

$Q_1$  - flow rate of input reactant ( $cm^3/sec$ );

$Q_2$  - flow rate of the recycled reactant ( $cm^3/sec$ );

$Q_3$  - flow rate of the liquid leaving the reactor ( $cm^3/sec$ );



$Q_4$  - flow rate of the product ( $cm^3/sec$ );  
 $Q_5$  - flow rate of cold water entering heat exchanger ( $cm^3/sec$ );  
 $C_a$  - concentration of reactant in the reactor;  
 $C_b$  - concentration of product in the reactor;  
 $C_{a0}$  - concentration of reactant in the input stream;  
 $r_a$  - reaction rate ( $g/sec$ );  
 $H_r$  - reaction heat constant ( $KJ/g$ );  
 $\rho$  - density of the reactant ( $g/cm^3$ );  
 $\rho_0$  - density of the solvent ( $g/cm^3$ );  
 $C$  - specific heat of the reactant ( $J/g^\circ C$ );  
 $C_0$  - specific heat of the solvent ( $J/g^\circ C$ );  
 $K_r$  - reaction rate constant ( $g/sec$ );  
 $a_r$  - constant peculiar to reaction ( $g/sec$ );  
 $b_r$  - constant peculiar to reaction ( $^\circ C$ );  
 $K_2$  - restriction parameter of valve 3 ( $cm^4/g^{1/2}sec$ );  
 $K_4$  - restriction parameter of valve 1 ( $cm^4/g^{1/2}sec$ );  
 $K_5$  - restriction parameter of valve 2 ( $cm^4/g^{1/2}sec$ );  
 $A_2$  - fractional opening of valve 3 (%);  
 $A_4$  - fractional opening of valve 1 (%);  
 $A_5$  - fractional opening of valve 2 (%);  
 $P$  - pressure of liquid leaving the pump ( $g/cm^2$ );  
 $P_0$  - pressure at the bottom of the reactor ( $g/cm^2$ );  
 $\Delta P$  - pressure increase caused by pump ( $g/cm^2$ );  
 $P_5$  - pressure of the feed cold water to the heat exchanger ( $g/cm^2$ );  
 $t$  - time ( $sec$ ).

The model parameters and the nominal values of certain process variables are given in the following table, Table 5.1. Therefore, from equations (5.9), (5.10) and (5.12), and using the model parameters values of  $A$ ,  $K_2$  and  $K_4$ , equation (5.1) can be simplified to,

$$\frac{dL}{dt} = \frac{Q_1 - 43.4A_4\sqrt{P}}{300} \quad (5.16)$$

Following a similar procedure, from equations (5.7) and (5.8) and taking the model parameters values of  $n$ ,  $a_r$ ,  $b_r$  and  $A$ , equation (5.2) can be represented as,

Parameters or variables	Values
$\rho$	1.2 g/cm <sup>3</sup>
$\rho_0$	1.1 g/cm <sup>3</sup>
$\Delta P$	200 g/cm <sup>2</sup>
$A$	300 cm <sup>2</sup>
$a_r$	0.8 g/sec
$b_r$	66.9 °C
$C$	0.9 J/g°C
$C_0$	0.8 J/g°C
$C_{a0}$	0.8
$H_r$	430 KJ/g
$K_2$	32.6 cm <sup>4</sup> /g <sup>1/2</sup> sec
$K_4$	43.4 cm <sup>4</sup> /g <sup>1/2</sup> sec
$K_5$	47 cm <sup>4</sup> /g <sup>1/2</sup> sec
$n$	1
$P_5$	200 g/cm <sup>2</sup>
$Q_1$	300 cm <sup>3</sup> /sec
$T_1$	20 °C
$T_5$	20 °C

Table 5.1 - Model parameters and nominal values of certain process variables

$$\frac{dC_a}{dt} = \frac{Q_1(C_{a0} - C_a) - 240C_aLe^{-66.9/T}}{300L} \quad (5.17)$$

and equation (5.3) can be reduced to,

$$\frac{dC_b}{dt} = \frac{240C_aLe^{-66.9/T} - C_bQ_1}{300L} \quad (5.18)$$

From equations (5.7), (5.8) and (5.9) and using the model parameters values of  $n$ ,  $a_r$ ,  $b_r$ ,  $A$ ,  $K_2$  and  $H_r$ , equation (5.4) can be simplified to,

$$\frac{dT}{dt} = \frac{B_1Q_1(T_1 - T) - 32.6A_2B_2\sqrt{P}(T - T_2) + 344 \times 10^3 C_a e^{-66.9/T}}{300LB_2} \quad (5.19)$$



Substituting equations (5.9) and (5.11) into equation (5.15), using the model parameters values of  $\rho$ ,  $\rho_0$ ,  $C$ ,  $C_0$ ,  $K_2$  and  $K_5$ , and using the process variables nominal values of  $T_5$  and  $P_5$ , gives,

$$T_2 = \frac{1169.8375A_5 + 3.26A_2\sqrt{P}T[1.08(C_a + C_b) + 0.88(1 - C_a - C_b)]}{58.4919A_5 + 3.26A_2\sqrt{P}[1.08(C_a + C_b) + 0.88(1 - C_a - C_b)]} \quad (5.20)$$

Substituting equation (5.14) into equation (5.13), and using the model parameters values of  $\rho$  and  $\rho_0$ , gives,

$$P = L[1.2(C_a + C_b) + 1.1(1 - C_a - C_b)] + \Delta P \quad (5.21)$$

Substituting the model parameters values of  $\rho$ ,  $\rho_0$ ,  $C$  and  $C_0$  into equations (5.5) and (5.6), gives respectively the following two equations,

$$B_1 = 1.08C_{a0} + 0.88(1 - C_{a0}) \quad (5.22)$$

$$B_2 = 1.08(C_a + C_b) + 0.88(1 - C_a - C_b) \quad (5.23)$$

Hence, briefly, the dynamic model of the CSTR system can be represented by the following set of equations, which are used to simulate the process.

$$B_1 = 1.08C_{a0} + 0.88(1 - C_{a0}) \quad (5.24)$$

$$B_2 = 1.08(C_a + C_b) + 0.88(1 - C_a - C_b) \quad (5.25)$$

$$T_2 = \frac{1169.8375A_5 + 3.26A_2\sqrt{P}TB_2}{58.4919A_5 + 3.26A_2\sqrt{P}B_2} \quad (5.26)$$

$$P = L[1.2(C_a + C_b) + 1.1(1 - C_a - C_b)] + \Delta P \quad (5.27)$$

$$\frac{dL}{dt} = \frac{Q_1 - 43.4A_4\sqrt{P}}{300} \quad (5.28)$$

$$\frac{dC_a}{dt} = \frac{Q_1(C_{a0} - C_a) - 240C_aLe^{-66.9/T}}{300L} \quad (5.29)$$

$$\frac{dC_b}{dt} = \frac{240C_aLe^{-66.9/T} - C_bQ_1}{300L} \quad (5.30)$$

$$\frac{dT}{dt} = \frac{B_1Q_1(T_1 - T) - 32.6A_2B_2\sqrt{P}(T - T_2) + 344 \times 10^3 C_a e^{-66.9/T}}{300LB_2} \quad (5.31)$$

Under normal operating conditions, the process variables  $Q_1$ ,  $\Delta P$ ,  $C_{a0}$  and  $T_1$  take the nominal values shown in Table 5.1, while in a fault simulation situation the values could be changed according to the fault or set of faults chosen by the operator. The process variables  $A_2$ ,  $A_4$  and  $A_5$  are controlled variables and, therefore, their values are the controllers outputs.

The qualitative model of the CSTR process is presented in the next sub chapter, which is derived from the dynamic model just achieved.

## 5.4 Qualitative Modelling of the Continuous Stirred Tank Reactor

A qualitative model is also developed for the continuous stirred tank reactor in a similar way as described in the previous chapter for the mixing process.

As this qualitative model will be used for fault detection purposes, it is assumed that the process is operating at a steady state prior to the occurrence of a fault or faults. Hence, the qualitative model for the continuous stirred tank reactor can be achieved based on its steady state model. Under this assumption, from equations (5.1) and (5.12), the following equation can be obtained,

$$Q_1 = Q_4 \quad (5.32)$$

Substituting equation (5.10) into equation (5.32), gives,

$$Q_1 = K_4 A_4 \sqrt{P} \quad (5.33)$$

From equations (5.13) and (5.14), equation (5.33) can be represented as,



$$Q_1 = K_4 A_4 \sqrt{L[\rho(C_a + C_b) + \rho_0(1 - C_a - C_b)] + \Delta P} \quad (5.34)$$

Considering the continuous stirred tank reactor in steady state conditions, equations (5.2) to (5.4) become,

$$0 = Q_1(C_{a0} - C_a) - r_a AL \quad (5.35)$$

$$0 = r_a AL - C_b Q_1 \quad (5.36)$$

$$0 = B_1 Q_1 (T_1 - T) - B_2 Q_2 (T - T_2) + H_r r_a \quad (5.37)$$

Equations (5.35) to (5.37) can also be represented as,

$$Q_1(C_{a0} - C_a) = r_a AL \quad (5.38)$$

$$Q_1 C_b = r_a AL \quad (5.39)$$

$$(B_1 Q_1 + B_2 Q_2)T = B_1 Q_1 T_1 + B_2 Q_2 T_2 + H_r r_a \quad (5.40)$$

Differentiating the two sides of equation (5.34), gives,

$$\begin{aligned} 2\sqrt{L[\rho(C_a + C_b) + \rho_0(1 - C_a - C_b)] + \Delta P} \frac{dQ_1}{dt} = \\ = K_4 A_4 [\rho(C_a + C_b) + \rho_0(1 - C_a - C_b)] \frac{dL}{dt} + \\ + K_4 \{L[\rho(C_a + C_b) + \rho_0(1 - C_a - C_b)] + \Delta P\} \frac{dA_4}{dt} + \\ + K_4 A_4 L(\rho - \rho_0) \frac{dC_a}{dt} + K_4 A_4 L(\rho - \rho_0) \frac{dC_b}{dt} + K_4 A_4 \frac{d\Delta P}{dt} \end{aligned} \quad (5.41)$$

Equation (5.41) can be re-formulated as,

$$\begin{aligned} K_4 A_4 [\rho(C_a + C_b) + \rho_0(1 - C_a - C_b)] \frac{dL}{dt} = \\ = 2\sqrt{L[\rho(C_a + C_b) + \rho_0(1 - C_a - C_b)] + \Delta P} \frac{dQ_1}{dt} - \\ - K_4 \{L[\rho(C_a + C_b) + \rho_0(1 - C_a - C_b)] + \Delta P\} \frac{dA_4}{dt} - \\ - K_4 A_4 L(\rho - \rho_0) \frac{dC_a}{dt} - K_4 A_4 L(\rho - \rho_0) \frac{dC_b}{dt} - K_4 A_4 \frac{d\Delta P}{dt} \end{aligned} \quad (5.42)$$

If, in equation (5.42), it is assumed that the changes in  $C_a$  and  $C_b$  cannot significantly affect the average density of the content in the reactor vessel and, hence, cannot significantly affect the pressure at the bottom of the reactor, then the above equation can be simplified to,

$$\begin{aligned}
 K_4 A_4 [\rho(C_a + C_b) + \rho_0(1 - C_a - C_b)] \frac{dL}{dt} = \\
 = 2\sqrt{L[\rho(C_a + C_b) + \rho_0(1 - C_a - C_b)] + \Delta P} \frac{dQ_1}{dt} - \\
 - K_4 \{L[\rho(C_a + C_b) + \rho_0(1 - C_a - C_b)] + \Delta P\} \frac{dA_4}{dt} - \\
 - K_4 A_4 \frac{d\Delta P}{dt}
 \end{aligned} \tag{5.43}$$

Taking the qualitative values of the two sides of equation (5.43) and using  $\Delta X$  to denote  $dX/dt$ , gives,

$$[\Delta L] = [\Delta Q_1] - [\Delta A_4] - [\Delta(\Delta P)] \tag{5.44}$$

Substituting equations (5.7) and (5.8) with  $n = 1$ , in equation (5.38), and then differentiating both sides, gives the following equation,

$$\begin{aligned}
 (C_{a0} - C_a) \frac{dQ_1}{dt} + Q_1 \frac{dC_{a0}}{dt} + Q_1 \frac{dC_a}{dt} = \\
 = Aa_r C_a e^{-b_r/T} \frac{dL}{dt} + Aa_r L e^{-b_r/T} \frac{dC_a}{dt} + \frac{Aa_r b_r L C_a e^{-b_r/T}}{T^2} \frac{dT}{dt}
 \end{aligned} \tag{5.45}$$

which can be re-formulated as,

$$\begin{aligned}
 \left( Aa_r L e^{-b_r/T} - Q_1 \right) \frac{dC_a}{dt} = \\
 = (C_{a0} - C_a) \frac{dQ_1}{dt} + Q_1 \frac{dC_{a0}}{dt} - Aa_r C_a e^{-b_r/T} \frac{dL}{dt} - \frac{Aa_r b_r L C_a e^{-b_r/T}}{T^2} \frac{dT}{dt}
 \end{aligned} \tag{5.46}$$

From equation (5.46) and following a similar procedure, as for the derivation of equation (5.43), gives the following qualitative algebraic equation,

$$[\Delta C_a] = [\Delta Q_1] + [\Delta C_{a0}] - [\Delta L] - [\Delta T] \tag{5.47}$$



Substituting equations (5.7) and (5.8) with  $n = 1$ , in equation (5.39), and then differentiating both sides, we achieve,

$$C_b \frac{dQ_1}{dt} + Q_1 \frac{dC_b}{dt} = Aa_r C_a e^{-b_r/T} \frac{dL}{dt} + Aa_r L e^{-b_r/T} \frac{dC_a}{dt} + \frac{Aa_r b_r L C_a e^{-b_r/T}}{T^2} \frac{dT}{dt} \quad (5.48)$$

which can be re-formulated as,

$$\begin{aligned} Q_1 \frac{dC_b}{dt} = & Aa_r C_a e^{-b_r/T} \frac{dL}{dt} + \frac{Aa_r b_r L C_a e^{-b_r/T}}{T^2} \frac{dT}{dt} + \\ & + Aa_r L e^{-b_r/T} \frac{dC_a}{dt} - C_b \frac{dQ_1}{dt} \end{aligned} \quad (5.49)$$

Taking the qualitative values of the two sides of equation (5.49), gives the following qualitative equation,

$$[\Delta C_b] = [\Delta L] + [\Delta T] + [\Delta C_a] - [\Delta Q_1] \quad (5.50)$$

Substituting equations (5.7) and (5.8) with  $n = 1$ , in equation (5.40), gives the following equation,

$$(B_1 Q_1 + B_2 Q_2) T = B_1 Q_1 T_1 + B_2 Q_2 T_2 + H_r a_r C_a e^{-b_r/T} \quad (5.51)$$

However, the variables  $B_1$  and  $B_2$  in equation (5.51) can approximately be treated as constants, if it is assumed that changes in  $C_{a0}$ ,  $C_a$  and  $C_b$  will not significantly affect the densities and specific heats of the input reactant and the content in the reactor vessel. Then, differentiating both sides of equation (5.51), gives the following equation,

$$\begin{aligned} B_1 T \frac{dQ_1}{dt} + B_2 T \frac{dQ_2}{dt} + (B_1 Q_1 + B_2 Q_2) \frac{dT}{dt} = \\ = B_1 T_1 \frac{dQ_1}{dt} + B_1 Q_1 \frac{dT_1}{dt} + B_2 T_2 \frac{dQ_2}{dt} + B_2 Q_2 \frac{dT_2}{dt} + \\ + \frac{H_r a_r b_r C_a e^{-b_r/T}}{T^2} \frac{dT}{dt} + H_r a_r e^{-b_r/T} \frac{dC_a}{dt} \end{aligned} \quad (5.52)$$

which can be re-formulated as,

$$\left( B_1 Q_1 + B_2 Q_2 - \frac{H_r a_r b_r C_a e^{-b_r/T}}{T^2} \right) \frac{dT}{dt} =$$

$$= B_1 Q_1 \frac{dT_1}{dt} + B_2 Q_2 \frac{dT_2}{dt} - B_1 (T - T_1) \frac{dQ_1}{dt} - B_2 (T - T_2) \frac{dQ_2}{dt} + H_r a_r e^{-b_r/T} \frac{dC_a}{dt}$$
(5.53)

Taking the qualitative values of the two sides of equation (5.53), gives,

$$[\Delta T] = [\Delta T_1] + [\Delta T_2] - [\Delta Q_1] - [\Delta Q_2] + [\Delta C_a]$$
(5.54)

The qualitative model for the continuous stirred tank reactor has now been achieved and is listed below,

$$[\Delta L] = [\Delta Q_1] - [\Delta A_4] - [\Delta(\Delta P)]$$
(5.55)

$$[\Delta C_a] = [\Delta Q_1] + [\Delta C_{a0}] - [\Delta L] - [\Delta T]$$
(5.56)

$$[\Delta C_b] = [\Delta L] + [\Delta T] + [\Delta C_a] - [\Delta Q_1]$$
(5.57)

$$[\Delta T] = [\Delta T_1] + [\Delta T_2] - [\Delta Q_1] - [\Delta Q_2] + [\Delta C_a]$$
(5.58)

This set of confluence's, which are formally derived from the quantitative equations for the process, will be used by a fuzzy qualitative simulation algorithm described in following chapters. In this way, the qualitative behaviour of physical variables  $L$ ,  $C_a$ ,  $C_b$  and  $T$  is predicted in the form of linguistic variables whose semantics are represented by fuzzy numbers. Moreover, the derivation of the qualitative model from the quantitative one ensures that the set of confluence's is consistent with the dynamic model of the process.

## 5.5 Process Variables Control

During simulation studies conducted with the continuous stirred tank reactor, control of some process variables has been performed. As quoted above, temperature and level in the reactor, as well as the recycle flow rate, are controlled by feed back control systems and classical controllers have been used.



To regulate level in the reactor a discrete PID controller is used of the form,

$$u(t) = K \left( e(t) + \frac{\sum_{i=1}^t e(i)}{T_i} + T_d [e(t) - e(t-1)] \right) \quad (5.59)$$

and the corresponding block diagram is depicted in Figure 5.3.

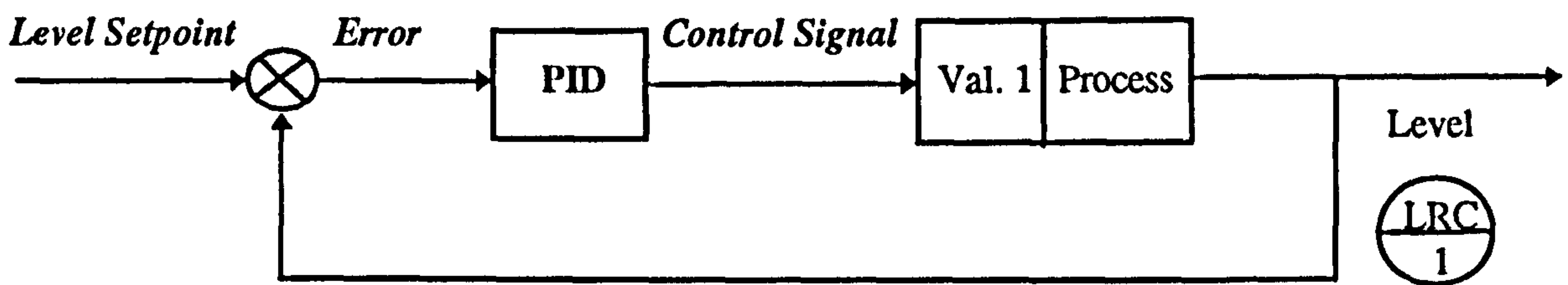


Figure 5.3 - Level control loop

On the other hand, to control temperature in the reactor a PI controller in cascade with a P controller is used and to control recycled flow rate a PI controller is used, which have respectively the following forms,

$$u(t) = K \left( e(t) + \frac{\sum_{i=1}^t e(i)}{T_i} \right) \quad (5.60)$$

$$u(t) = Ke(t) \quad (5.61)$$

The corresponding block diagrams of the control loops are depicted in Figures 5.4 and 5.5, respectively.

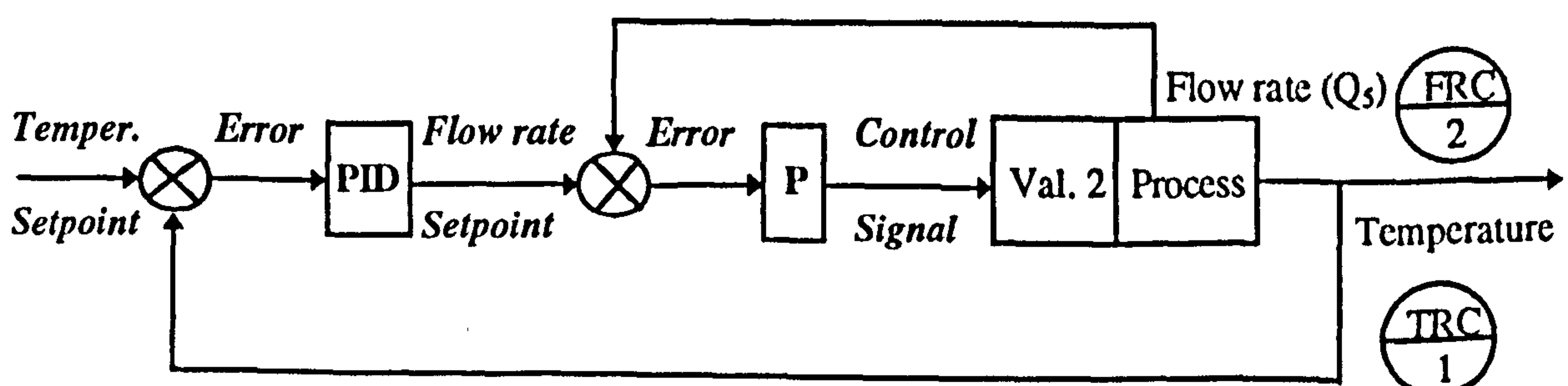


Figure 5.4 - Temperature control loop

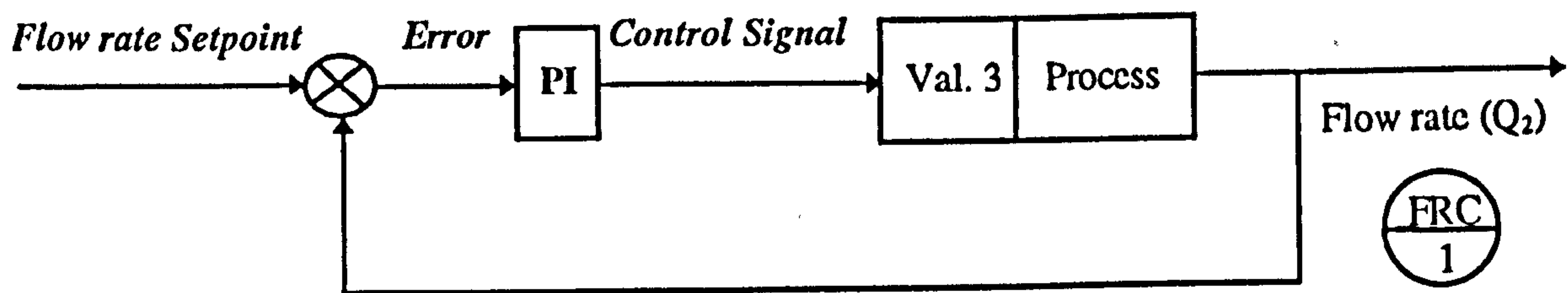


Figure 5.5 - Recycled flow rate control loop

In the control laws defined by equations (5.59), (5.60) and (5.61) the following notation is used:

- $u(t)$ , stands for control signal at time instant  $t$ ;
- $e(t)$ , stands for error signal at time instant  $t$ ;
- $K$ , is the controller gain;
- $T_i$ , is the integration time;
- $T_d$ , is the derivative time;
- $t$ , stands for sampling time instant.

The controllers performance is discussed in the next sub chapter, where some results achieved during simulation studies conducted with the continuous stirred tank reactor, are also presented.

## 5.6 Controllers Performance

In order to perform the process simulation a fifth-order Runge-Kutta method is used to solve the set of differential equations which constitutes the dynamic model of the process. For this simulation a sampling time interval of 1 second is used, and the corresponding parameters values of the controllers are given in Table 5.2.

Control Loop	Controllers Parameters		
	$K$	$T_i$	$T_d$
$L$	0.08	20.0	2.86
$Q_2$	0.1	5.0	-----
$T$	8.0*	1.2*	-----
	0.02**	-----	-----

\* Primary control loop

\*\* Secondary control loop

Table 5.2 - Controllers parameters



During the simulation studies performed with the continuous stirred tank reactor, a satisfactory controls performance has been observed. An example can be seen in Figure 5.6, where the response of the PID controller for the level control loop and the response of the PI controller for the recycled flow rate are shown. A low overshoot, as well as a low undershoot, for both control loops is observed.

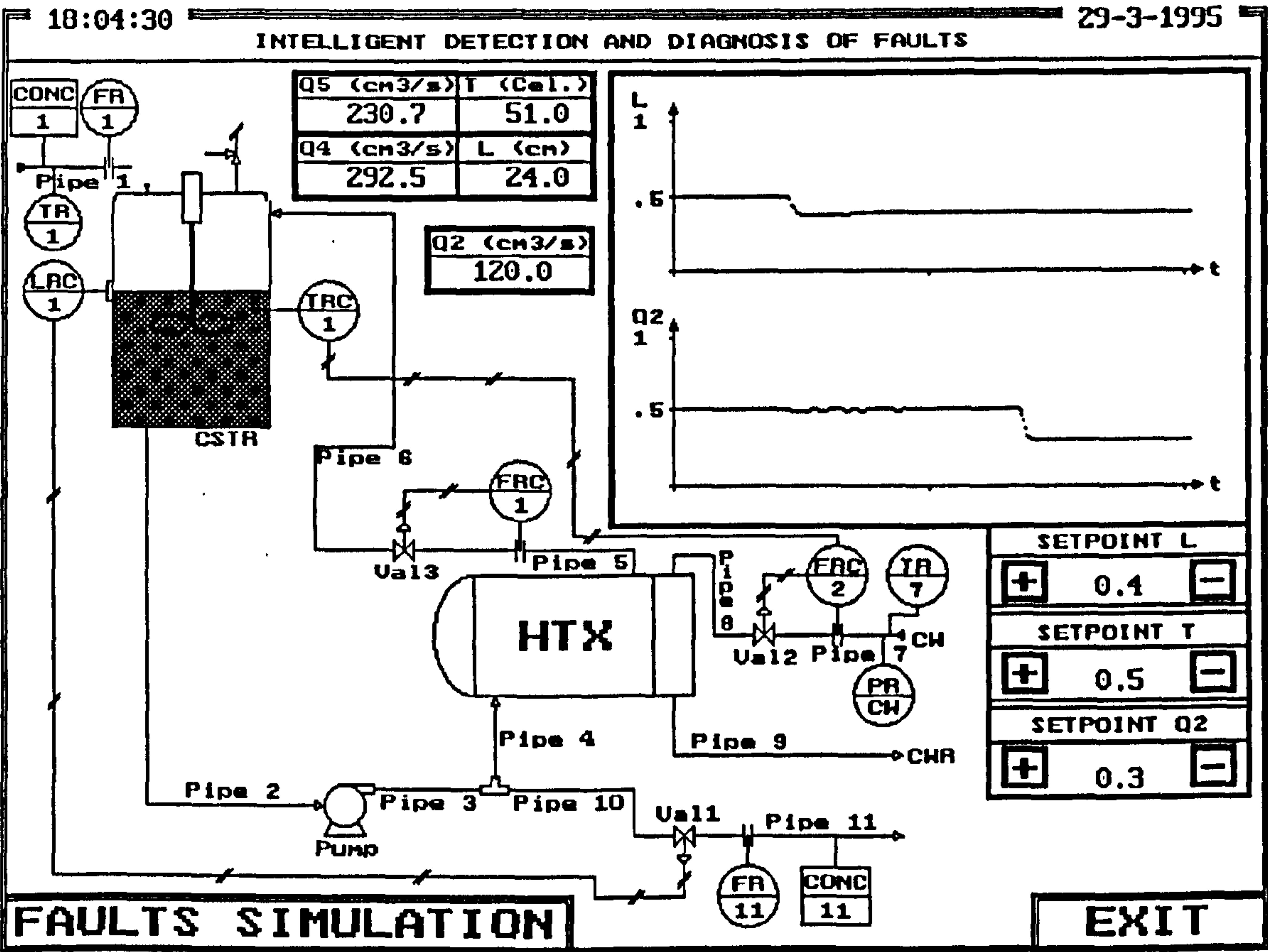


Figure 5.6 - Controllers performance

### 5.7 Conclusions

In this chapter a continuous stirred tank reactor is presented. For such a process a dynamic mathematical model is derived in sub chapter 5.3. Moreover, a TURBO C++ program has been implemented to perform the process simulation. During simulation studies conducted with the computational system developed, the dynamic model of the process has been used to achieve the process simulation. In further chapters this model will be used to simulate the process under normal operating conditions, as well as under fault or faults situations. This will allow the performance and reliability of several fault detection and diagnosis systems presented in the following chapters to be tested.

From the dynamic mathematical model of the continuous stirred tank reactor a qualitative model, which is represented by a set of confluence's has been obtained. The qualitative model of the process will be used in further chapters for fault detection purposes in the process under consideration.

Control of some process variables has been performed through conventional controllers. The results, which have been obtained during simulation studies, have shown a good performance of the control system implemented.



## Chapter 6

# On-Line Single Fault Diagnosis Based on Fuzzy Qualitative Simulation

### 6.1 Introduction

During the last two decades, the so called model based fault diagnosis approach has received increasing attention in both research and application, providing plant operators with a complete knowledge of the plant's operating state in order to maximise safety, efficiency and quality of operation (Frank 1990 and Patton *et al.* 1989). Accurate and timely information enables operators to respond rapidly to plant disturbances and minimise the effects of system failures on the plant, product and the environment. The early detection and identification of degradation or impending failure of a plant component or sensor can provide vital information to the plant operator to assist in the task of controlling the plant.

The goal of early fault detection is to recognise the occurrence of a fault early enough in order to permit the control system to resume normal operation or operate with degraded but acceptable performance or initiate action for a controlled shut down. The traditional techniques of fault diagnosis involve limit value checking of some process variables or simple plausibility checks. Clearly, this method is not capable of performing a high quality fault diagnosis, because it can not provide much information about the locations and sources of faults. Moreover, with current monitoring technology, alarms are triggered whenever fixed threshold values are exceeded. In process plants with complex interactions and tight coupling, hundreds or thousands of distinct alarms can be activated within a minute. In such situations, process operators tend to overlook relevant information, respond too slowly, and panic when the rate of information flow is too great.



For these reasons there is motivation to develop a computational system consisting of a process simulator with control of some process variables in a lower layer and a supervisory module in an upper layer whose main aim is to detect and diagnose faults introduced in the lower layer. The fault detection and diagnosis system implemented has the ability to detect and diagnose single abrupt faults, and has been applied during simulation studies conducted with the mixing process which has been described at chapter 4.. The expert system developed is based on Artificial Intelligence techniques and is described below.

Depending on the depth of the process knowledge employed, the techniques, applied in the development of the most recent detection and diagnosis systems, can be classified into quantitative and qualitative approaches. Quantitative diagnostic systems utilise a rigorous process model and on-line measurements to back-calculate the crucial process variables (Willsky 1976, Isermann 1984). A process fault is characterised by the significant deviation in the calculated process variables. This approach requires extensive quantitative process knowledge, to perform computation quantitatively. However, for some processes, accurate model parameters may not be available, and in some cases, accurate or direct measurements of some process variables may also be unavailable. So, qualitative information may be more adequate for fault diagnosis and this is the basis of the present computational system. Moreover, the use of qualitative information allows an increase in the computational speed, which is fundamental for an early fault detection.

The qualitative approach, require much less process knowledge. This method is based upon the concept of a qualitative model which unlike the quantitative approach only requires declarative information, such as, the sign of variables, the tendencies of variables increasing (+), decreasing (-) or steady (0) together with relative magnitude. Indeed qualitative models result in diagnostic systems that are inherently more robust than numerically based systems. One advantage of this method is that the effect of a fault can be easily represented by the deviation of the corresponding process variables and, hence, the qualitative model can easily be used to simulate the process under normal or various faulty conditions. This property is essential in monitoring a physical system, whether healthy or faulty.

Studies of qualitative modelling are currently being conducted in the field of artificial intelligence. Previous methods of qualitative modelling have tended to suffer from excessive generation of multiple solutions, which could lead to a loss in diagnostic resolution if these methods were used in practice. Spurious solutions which may be generated when there are competing qualitative influences which cannot be resolved, for instance when one parameter tends to make a variable increase, and another tends to make it decrease. In general, qualitative simulation cannot be guaranteed to exclude spurious solutions (Kuipers 1986); however, several strategies for reducing spurious solutions and/or ambiguity have been pursued.



There are several different methods in qualitative reasoning such as de Kleer and Brown's confluence based qualitative reasoning (de Kleer and Brown 1984) which is also referred to as Incremental Qualitative Analysis (IQA)(Herbert and Williams 1987), Forbus' qualitative process theory (Forbus 1984), Kuipers' qualitative simulation (Kuipers 1986, 1987), Kramer and Palowitch's signed directed graph (Kramer and Palowitch 1987), and Shen and Leitch's fuzzy qualitative simulation (Shen and Leitch 1993).

However, due to the lack of quantitative information, ambiguity often occurs in qualitative reasoning, especially when a large number of qualitative variables are involved. Several methodologies have been pointed out to minimise the effects of ambiguity in qualitative reasoning. For instance, in order to solve this problem, Zhang (1991) used an extended method of order of magnitude reasoning proposed by Raiman (1986). In the current approach the theory of fuzzy sets has been used to overcome this problem and increase the system's reliability.

The qualitative model used in this approach is a set of confluence's which are qualitative equations and are derived from a quantitative model of the process under consideration, following a similar procedure as proposed by de Kleer and Brown (1984). The approach presented by the author utilises the theory of fuzzy set to give an arbitrary, but finite, discretisation of the representation of system variables (Dubois and Prade 1980). Linguistic variables defined on the interval,  $[-1, 1]$ , are interpreted as verbal probabilities and their semantics are represented by fuzzy numbers. The term set of linguistic variables defines the granularity of the confidence assessment values that can consistently be expressed by users or experts. Moreover, the adoption of fuzzy sets allows common-sense knowledge to be incorporated in the interpretation of values through the use of graded membership. Thus, fuzzy heuristic rules can be used to perform this task. Such a system also allows both magnitude and sign information on the functional relationship holding-against two or more variables to be represented, resulting in a considerable reduction of the inherent ambiguity of qualitative computation.

Such an approach uses a qualitative level of description that lets us express imprecise knowledge and takes advantage of quantitative knowledge when it is available, which is usually the case in process plants. Moreover, it permits direct comparison of the numeric sensor readings transformed into linguistic variables, whose semantics are represented by fuzzy numbers, with the linguistic values predicted for each variable.

In the next sub chapter, a description of the computational system architecture implemented is presented. Sub chapter 6.3 describes the fuzzy qualitative reasoning, which has been used as a first stage of a fault detection. Sub chapter 6.4 presents the methodology used to detect a fault in the process under concern. The fault diagnosis reasoning is described in sub chapter 6.5. The application of such a fault detection and diagnosis system to the simulated mixing



process, presented at chapter 4., is described in sub chapter 6.6. In this sub chapter some results, achieved during simulation studies, are also presented. The last sub chapter contains some concluding remarks.

## 6.2 System Architecture

This sub chapter presents a general description of the computational system implemented. The mixing process presented in the previous chapter has been used as a test bed for the fault detection and diagnosis system developed. Therefore, control of level and temperature of tank 2 has been considered, which has been performed through the rule based controller described in chapter 4.. The overall computational system is implemented in Turbo C++ and run in a personal computer without special features. For instance during the current research a 386 (25 MHz) machine, fitted with a mathematical co-processor, has been used. A block diagram of such a computational system is depicted in Figure 6.1.

This computational system can be interpreted as a system developed in two layers. In the lower layer we have the process simulator and the respective control. The upper layer consists of a supervisor system, whose main goal is to detect and diagnose faults simulated in the lower layer. The module identified by Data Fuzzification makes the interface between the two layers, which samples the process variables and transforms their values into linguistic variables whose semantics are represented by fuzzy numbers.

To perform the process simulation in the lower layer the dynamic model of the process under consideration is used in the form of differential equations. For instance, for the mixing process the simulation has been achieved through the dynamic model represented by equations (4.13) to (4.18). To solve this set of ordinary differential equations a fifth-order Runge-Kutta method is used. In order to obtain a predetermined accuracy in the solution with minimum computational effort, an adaptive stepsize for the Runge-Kutta method is utilised. So, the integrator of the ordinary differential equations set exert some adaptive control over its progress making frequent changes in its stepsize.

As quoted above the data fuzzification block performs the interface between the layers that constitute the system, process simulator/control and fault detection/diagnosis. The process variables are sampled with a sampling rate of 5 seconds and transformed into linguistic variables whose semantics are represented by fuzzy numbers. For simplicity, however, the author has used a normalised range  $[-1, 1]$  to form the basis on which the fuzzy quantity space is discretized. The following fuzzy quantity space was adopted,  $q_f = \{nlarge, nmedium, nsmall, zero, psmall,$



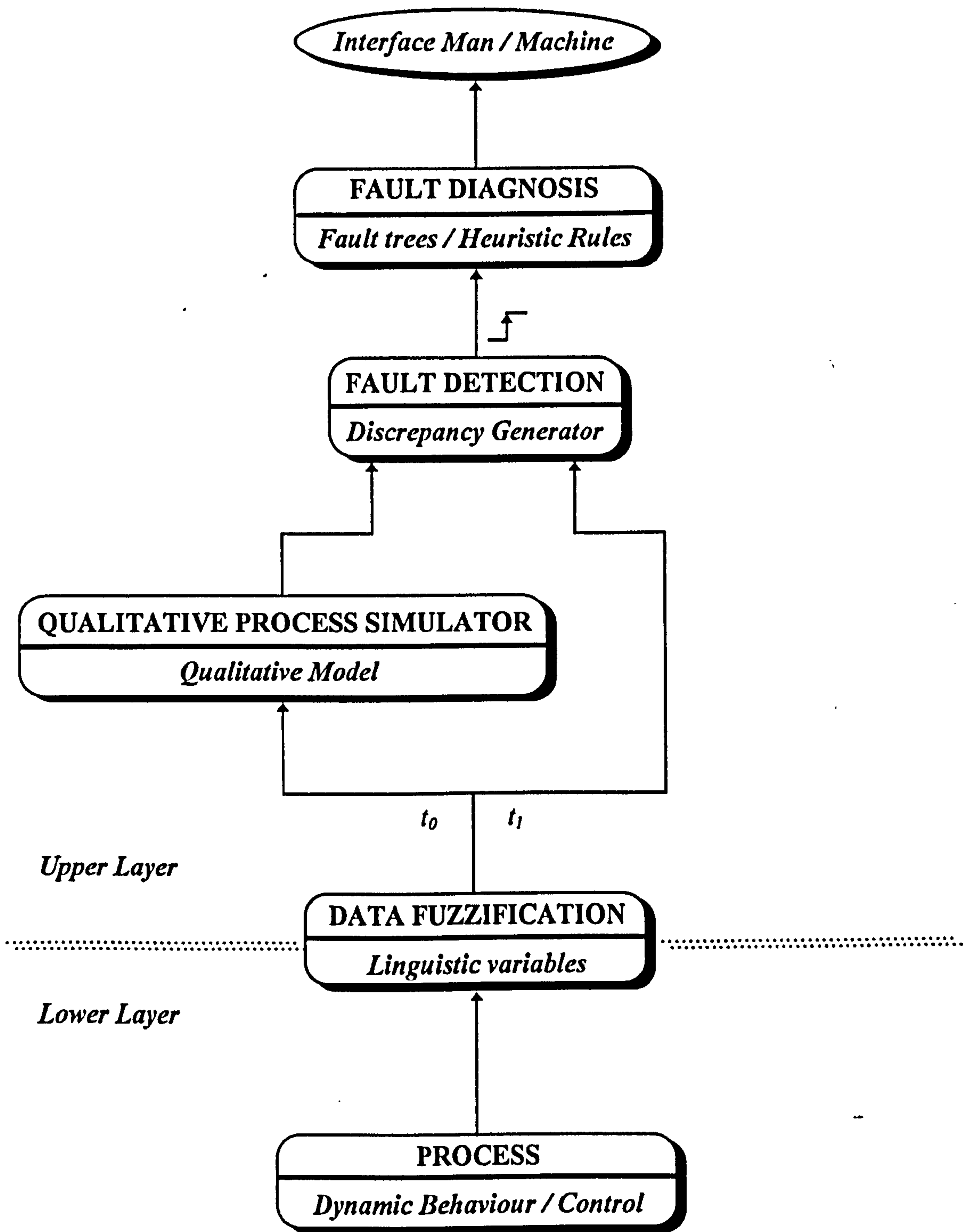


Figure 6.1 - General scheme of a computer assisted fault diagnosis system

$p_{medium}, p_{large}\}$ , with the qualitative values being represented by seven 4-tuple parametric fuzzy numbers as given below and shown in Figure 6.2,

$$qf = \{[-1, -0.75, 0, 0.1], [-0.6, -0.45, 0.1, 0.15], [-0.3, -0.1, 0.05, 0.1], [0, 0, 0, 0], \\ [0.1, 0.3, 0.1, 0.05], [0.45, 0.6, 0.15, 0.1], [0.75, 1, 0.1, 0]\}$$

The corresponding membership functions, of such seven 4-tuple fuzzy numbers, are depicted in Figure 6.2.

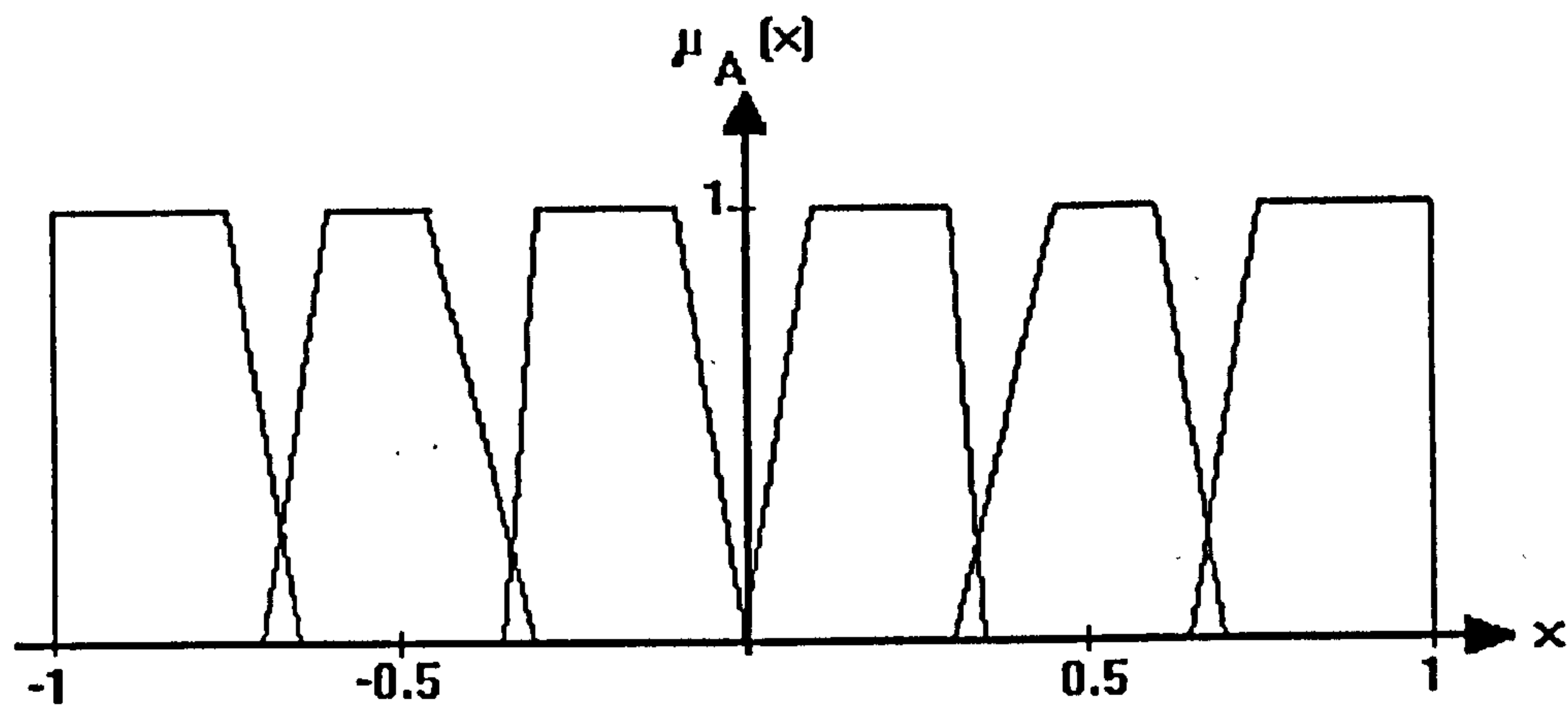


Figure 6.2 - The representation of the fuzzy quantity space

However, in a general situation, design engineers usually cannot determine membership functions of a fuzzy number precisely (Yagawa *et al.* 1991). Therefore, the membership functions need certain tuning processes before their utilisation. In the present study, the fuzzy membership functions are first tuned up through a trial and error procedure.

The information handled by the diagnosis system is in qualitative form which is converted from on-line quantitative information. To perform this task it is necessary to define a parameter for each variable which makes the normalisation of the different changes in the process variables in the range  $[-1, 1]$ . However, since the fault detection system is triggered only if a sample process variable presents a change greater than "*psmall*" or less than "*nsmall*", the normalisation parameters will affect the performance of the detection system and should be set properly. Large normalisation parameters could make the fault detection system very sensitive to process disturbances as well as in a real situation very sensitive to measurement noise, and may result in spurious triggers of the fault detection system. Small normalisation parameters may miss faults. During the current studies, it is found that the proper setting of these normalisation parameters, used to normalise the changes in the process variables, can remarkably save computational time and increase the performance of the system.

These parameters are set based on previous operational experience of the process under concern. To set the normalisation parameters of each sampled process variable the author performed the following procedure:

1. Random changes in the magnitudes of the setpoints of the temperature as well as of the level of tank 2 are performed. The consequent changes in the sampled variables are collected in a file with ASCII format.



2. This file is imported by a spreadsheet such as Lotus or Excel and with the data collected corresponding flowcharts are achieved. A example of the flowchart achieved for level in tank 1 is shown in Figure 6.3.
3. The normalisation parameter is achieved such that the most part of variable changes is converted into a linguistic variable "*psmall*" or "*nsmall*". For the example shown in Figure 4.3, the normalisation parameter 1 is obtained.
4. Through a trial and error procedure a fine adjustment of some normalisation parameters is performed in order to obtain the desirable results in qualitative simulation.

This procedure allows qualitative simulation to be performed through arithmetic operations with fuzzy numbers that correspond to changes in process variables with different physical meaning.

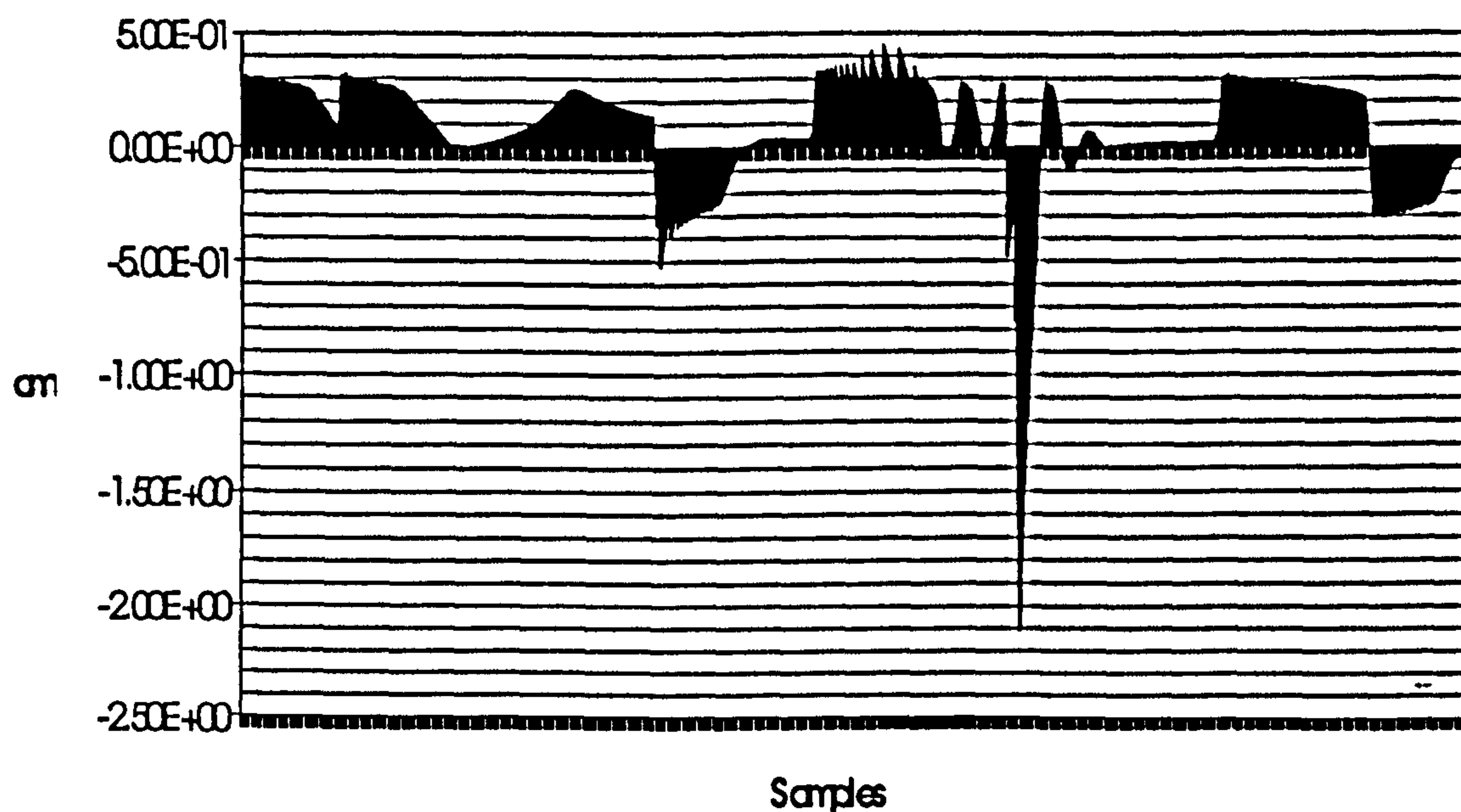


Figure 6.3 - Changes in level of tank 1

However, if any normalised change of a process variable, has a value greater than 1 or less than -1, the linguistic variables "*plarge*" and "*nlarge*" will be assigned accordingly. For instance, since the normalisation parameter for level of tank 1 takes the value 1, some situation such that seen in Figure 6.3 occurs where the changes in the level of tank 1 take values less than -1 cm. Therefore, in such a situation, the linguistic value "*nlarge*" will be assigned.

The main parts of the fault detection and diagnosis system, such as the fuzzy qualitative simulation as well as the fault detection and diagnosis reasoning, are described in the next sub chapters.



## 6.3 Fuzzy Qualitative Reasoning

The aim of the Qualitative Process Simulator block is to predict the behaviour of the measured process variables under normal operational conditions as well as under fault conditions, based on a qualitative reasoning approach. The qualitative reasoning method is specially appropriate when the monitored process contains a large number of variables and also because it depends less on accurate quantitative information. This is particularly useful in simulating the effect of a fault as the exact severity of a fault is generally not known. Moreover, in contrast with quantitative approaches, noise in the measurement variables has a reduced effect in qualitative reasoning.

Typical methods of qualitative simulation for the description of a physical system are based on initial use of quantitative knowledge, and normally only sign information on the rate of change of each variables is represented, lacking ordering information amongst the rates of change. In the present approach a new methodology is performed based on the use of linguistic variables which semantics are represented by fuzzy numbers.

Thus, due to the lack of quantitative information, ambiguity often occurs in qualitative reasoning, especially when a large number of qualitative variables are involved. In order to reduce the ambiguity, the sampled process variables are converted into fuzzy numbers by a fuzzification system. Therefore, in some sense, we can say that this is a system based on a qualitative/quantitative approach. However, the use of fuzzy numbers allows the introduction of common sense knowledge and by this means the ambiguity can be eliminated.

As quoted above several different approaches in qualitative reasoning are pointed out by several researchers. The qualitative model used in this approach is a set of confluence's which are qualitative equations derived from a quantitative model of the process under concern. The qualitative simulation is triggered when at least one of the sampled process variables takes one of the following linguistic variables: *pmedium*; *plarge*; *nmedium*; *nlarge*. Here the aim is to predict the behaviour of the process variables, in order to confirm or to deny a hypothetical fault, which is performed by the fault detection module. The sampled process variables, converted into linguistic variables by the data fuzzification system, are propagated through the set of confluence's. Following this procedure, according to the qualitative model of the process under consideration, the rate of change of some process variables can be predicted in a qualitative form, such that the semantics of the results are represented by fuzzy numbers.

Therefore, as the semantics of the linguistic variables are represented by 4-tuple fuzzy numbers, their propagation through the qualitative model is performed using expressions (3.19) and (3.20). However, in order to avoid computational difficulties the arithmetic operations with fuzzy



numbers have been implemented in the form of a matrix as shown in Tables 6.1 and 6.2, respectively for addition and subtraction.

+	<i>nlarge</i>	<i>nmedium</i>	<i>nsmall</i>	<i>zero</i>	<i>psmall</i>	<i>pmedium</i>	<i>plarge</i>
<i>nlarge</i>	<i>nlarge</i>	<i>nlarge</i>	<i>nlarge</i>	<i>nlarge</i>	<i>nmedium</i>	<i>nsmall</i>	<i>nsmall or psmall</i>
<i>nmedium</i>	<i>nlarge</i>	<i>nlarge</i>	<i>nlarge</i>	<i>nmedium</i>	<i>nsmall</i>	<i>nsmall or psmall</i>	<i>psmall</i>
<i>nsmall</i>	<i>nlarge</i>	<i>nlarge</i>	<i>nmedium</i>	<i>nsmall</i>	<i>nsmall or psmall</i>	<i>psmall</i>	<i>pmedium</i>
<i>zero</i>	<i>nlarge</i>	<i>nmedium</i>	<i>nsmall</i>	<i>zero</i>	<i>psmall</i>	<i>pmedium</i>	<i>plarge</i>
<i>psmall</i>	<i>nmedium</i>	<i>nsmall</i>	<i>nsmall or psmall</i>	<i>psmall</i>	<i>pmedium</i>	<i>plarge</i>	<i>plarge</i>
<i>pmedium</i>	<i>nsmall</i>	<i>nsmall or psmall</i>	<i>psmall</i>	<i>pmedium</i>	<i>plarge</i>	<i>plarge</i>	<i>plarge</i>
<i>plarge</i>	<i>nsmall or psmall</i>	<i>psmall</i>	<i>pmedium</i>	<i>plarge</i>	<i>plarge</i>	<i>plarge</i>	<i>plarge</i>

Table 6.1 - Addition of fuzzy numbers

-	<i>nlarge</i>	<i>nmedium</i>	<i>nsmall</i>	<i>zero</i>	<i>psmall</i>	<i>pmedium</i>	<i>plarge</i>
<i>nlarge</i>	<i>nsmall or psmall</i>	<i>nsmall</i>	<i>nmedium</i>	<i>nlarge</i>	<i>nlarge</i>	<i>nlarge</i>	<i>nlarge</i>
<i>nmedium</i>	<i>psmall</i>	<i>nsmall or psmall</i>	<i>nsmall</i>	<i>nmedium</i>	<i>nlarge</i>	<i>nlarge</i>	<i>nlarge</i>
<i>nsmall</i>	<i>pmedium</i>	<i>psmall</i>	<i>nsmall or psmall</i>	<i>nsmall</i>	<i>nmedium</i>	<i>nlarge</i>	<i>nlarge</i>
<i>zero</i>	<i>plarge</i>	<i>pmedium</i>	<i>psmall</i>	<i>zero</i>	<i>nsmall</i>	<i>nmedium</i>	<i>nlarge</i>
<i>psmall</i>	<i>plarge</i>	<i>plarge</i>	<i>pmedium</i>	<i>psmall</i>	<i>nsmall or psmall</i>	<i>nsmall</i>	<i>nmedium</i>
<i>pmedium</i>	<i>plarge</i>	<i>plarge</i>	<i>plarge</i>	<i>pmedium</i>	<i>psmall</i>	<i>nsmall or psmall</i>	<i>nsmall</i>
<i>plarge</i>	<i>plarge</i>	<i>plarge</i>	<i>plarge</i>	<i>plarge</i>	<i>pmedium</i>	<i>psmall</i>	<i>nsmall or psmall</i>

Table 6.2 - Subtraction of fuzzy numbers

In order to illustrate the importance of the distance measured principle quoted in chapter 3., in the qualitative reasoning, the follow example is presented. Thus, let us consider the confluence for the mixing process, represented by the following equation,

$$[\Delta L_2] = [\Delta Q_{o1}] - [\Delta Q_{o2}]$$
(6.1)

where  $[\Delta Q_{01}]$  takes the qualitative value *psmall* while  $[\Delta Q_{02}]$  takes the qualitative value *pmedium*. As the semantics of *psmall* and *pmedium* are represented by 4-tuple fuzzy numbers,  $[0.1, 0.3, 0.1, 0.05]$  and  $[0.45, 0.6, 0.15, 0.1]$  respectively, we have,

$$[\Delta \dot{L}_2] = [0.1, 0.3, 0.1, 0.05] - [0.45, 0.6, 0.15, 0.1] \quad (6.2)$$

Then, according to the expression (3.20), the result of expression (6.2) is the following 4-tuple fuzzy number,

$$[\Delta \dot{L}_2] = [-0.5, -0.15, 0.2, 0.2] \quad (6.3)$$

By checking if an element in  $q_f$  intersects with  $\hat{X} = [-0.5, -0.15, 0.2, 0.2]$ , the following set of 4-tuple fuzzy numbers, is generated,

$$_{\hat{X}} q_f = \{[-0.6, -0.45, 0.1, 0.15], [-0.3, -0.1, 0.05, 0.1], [0, 0, 0, 0], [0.1, 0.3, 0.1, 0.05]\} \quad (6.4)$$

This implies that  $[-0.6, -0.45, 0.1, 0.15], [-0.3, -0.1, 0.05, 0.1], [0, 0, 0, 0]$  and  $[0.1, 0.3, 0.1, 0.05]$  are fuzzy values that  $[\Delta \dot{L}_2]$  may take. However, the utilisation of the approximation principle allows a distinction to be made between the possible values of a variable. In fact, distances between  $\hat{X}$  and  $X$ , where  $X \in _{\hat{X}} q_f$ , can be evaluated through the expressions (3.22), (3.23) and (3.24), and the results are,

$$d = \{0.340, 0.302, 0.639, 0.593\} \quad (6.5)$$

Therefore, the approximation of  $[-0.5, -0.15, 0.2, 0.2]$  is deemed to be  $[-0.3, -0.1, 0.05, 0.1]$ , based on the smaller distance 0.302. This results in,

$$[\Delta \dot{L}_2] = [-0.3, -0.1, 0.05, 0.1] = nsmall \quad (6.6)$$

So, we have,

$$psmall - pmedium = nsmall \quad (6.7)$$



as shown in the Table 6.2, and is well suited to our common sense calculus, putting the problem of order of magnitude reasoning on a firm basis. This simple example clearly shows that the ambiguity with regard to conventional sign algebra (de Kleer and Brown 1984) is significantly reduced with an extended quantity space.

Even in the case where differences between the distances are not obtained, the calculations can result in only two values. For instance, let us consider the following,

$$plarge + nlarge = [-0.25,0.25,0.1,0.1] \tag{6.8}$$

Now, by checking if an element in  $q_f$  intersects with  $\hat{X} = [-0.25,0.25,0.1,0.1]$ , the following set is generated,

$${}_Xq_f = \{[-0.3,-0.1,0.05,0.1],[0,0,0,0],[0.1,0.3,0.1,0.05]\} \tag{6.9}$$

In this case, the calculation of the distances between  $\hat{X}$  and  $X$ , where  $X \in {}_Xq_f$ , gives the following result,

$$d = \{0.382,0.632,0.382\} \tag{6.10}$$

Therefore, as shown in Table 6.1, the result of expression (6.8) can be *nsmall* or *psmall*. However, in this situation, the fact that we are using fuzzy numbers to perform the qualitative simulation, allows common sense knowledge in the form of heuristic rules, which can eliminate these spurious solutions, to be used.

For instance, during the simulation studies conducted with the mixing process, when we

Process Variables	Predicted Behaviour
L <sub>1</sub>	Increase - plarge
L <sub>2</sub>	Decrease - nsmall or Increase - psmall
T <sub>1</sub>	Steady - zero
T <sub>2</sub>	Steady - zero

Table 6.3 - Predicted behaviour of process variables under "Hand Valve 1 is blocked" fault situation.

initiate the fault situations of “Hand Valve 1 blocked” and “Hand Valve 2 blocked” with the steady state condition, setpoints of the level and temperature in tank 2 equal to 50%, the predicted behaviour of level and temperature in both tanks is presented in Table 6.3 and Table 6.4 respectively. It can be seen that there are two situations with spurious solutions. Fortunately, the use of common sense knowledge can solve this ambiguity problem.

Process Variables	Predicted Behaviour
$L_1$	Increase - psmall
$L_2$	Decrease - nlarge
$T_1$	Decrease - nsmall or Increase - psmall
$T_2$	Steady - zero

Table 6.4 - Predicted behaviour of process variables under "Hand Valve 2 is blocked" fault situation.

Therefore, in order to solve this ambiguity problems the following three heuristic rules can be introduced in the knowledge base,

1. IF ( $[\Delta \dot{L}_2] = \text{indeterminate AND } Q_{o1} = 0$ ) THEN  $[\Delta \dot{L}_2] = \text{nsmall};$
2. IF ( $[\Delta \dot{L}_2] = \text{indeterminate AND } Q_{o2} = 0$ ) THEN  $[\Delta \dot{L}_2] = \text{psmall};$
3. IF ( $[\Delta Q_h] = [\Delta Q_c]$ ) THEN  $[\Delta \dot{T}_1] = \text{zero}.$

Looking at the qualitative model of the mixing process, which is presented in chapter 4., the first and second rules are obvious while the third rule can be obtained from the controller mechanism explained in sub chapter 4.5. It is worth noting that the predicted behaviour of the level in the tank 2, presented in Table 6.4, appears strange. However, we obtained this prediction because the fault is confirmed only in the second time that the qualitative simulator is triggered after fault initialisation. The used of this specific knowledge about the process under concern could increase the reliability of the fault detection system.

The results of the qualitative simulation are passed to the fault detection block where they are compared with the real behaviour of the process variables in the next sampled time. In the next sub chapter the fault detection reasoning is discussed.



## 6.4 Fault Detection Reasoning

Fault detection, which is closely related to monitoring, involves differentiating between normal and abnormal conditions. Managing this kind of problem solving requires reasoning about physical relationships in a way that explains the current process state and predicts trajectories that the process is likely to follow.

There are two ways to view the fault detection problem. In the first view past experience plays the dominant role. Experienced failure situations are coded as heuristic rules, perhaps together with some predictive or statistical knowledge, all obtained from a human expert in a particular domain. This could, in some sense, be considered as the traditional expert system approach. However, the trial and error process by which knowledge is elicited, programmed, and tested is likely to produce inconsistent and incomplete databases and, hence, an expert system may exhibit important gaps in knowledge at unexpected times (Denning 1986). Obviously, such situations can have serious consequences in some process industries.

The second view, and the view taken in the approach presented here, is a fault detection system based on a functional model of some system and some measurements from that system. If the measurements conflict with the functional model then there is a diagnostic problem, which is dealt with as presented in sub chapter 6.5. This model based approach to fault detection and diagnosis has emerged from two different communities. In the engineering community, fault detection and diagnosis techniques generally rely on a precise mathematical model of the process and on pre-enumerated fault symptom patterns known as fault signatures. In the computer science/artificial intelligence community, model based fault detection and diagnosis systems rely on models of structure and behaviour.

The process plant knowledge used here is based on general laws of physics and chemistry; that is, a description of the physical and chemical laws that the plant obeys. The method of description is qualitative physics, which attempts to model the plant in terms of a model of the general physical and chemical relationships in the plant rather than by rigorous mathematical modelling of the process. The current model based approach has evolved within the artificial intelligence community.

The detection of a fault is based on the comparison of the predicted behaviour with the real behaviour of the process variables. Thus, when the qualitative simulator system is triggered the behaviour of some process variables are predicted following the procedure described in the previous sub chapter, and the results are passed to the fault detection block. Then, in time instant  $t+1$ , the data fuzzification system samples the real behaviour of the process variables whose behaviour has already been predicted, and converts these values to linguistic variables whose



semantics are also represented by fuzzy numbers. The linguistic variables achieved in this way are then passed to the fault detection system. The next step of the procedure is to compare the results of the qualitative simulator system with the real behaviour of the process variables. Then if the real behaviour of at least one process variable does not match the linguistic value predicted by the qualitative simulator, the existence of a fault in the process under concern is confirmed. Otherwise, the existence of a fault is denied.

Hence, the fault detection system can be called a discrepancy generator. This means that when any discrepancy is achieved the existence of a fault is confirmed, and the fault diagnosis system is triggered in order to locate the fault or faults. Otherwise the process simulation continues until the qualitative simulator system is triggered with a fault confirmed by the fault detection module. The procedure used to locate a fault in the process under consideration is described in the next sub chapter.

## **6.5 Fault Diagnosis Reasoning**

Malfunction diagnosis isolates and identifies process malfunctions. This can be especially difficult in large malfunction hypothesis spaces. Navigating these hypothesis spaces requires that we structure diagnostic knowledge in forms that are efficient for problem solving. Therefore, since search forms the core of the fault diagnosis system, it is useful to structure the program in a way that facilitates the search process.

Hence, according to chapter 2. the fault diagnosis system has been implemented as a production system, which has been shown to be a good way to model the strong data-driven nature of intelligent actions. Moreover, the object of a search procedure is to discover a path through a problem space from an initial configuration to a goal state. As discussed earlier there are two directions in which such a search could proceed:

- Forward, from the start states;
- Backward, from the goal states.

In the system developed, both techniques are used in the diagnostic process. A general analysis is carried out to find what faults are possible and generate a reduced set of fault candidates; forward chaining. A diagnosis is then selected and a more detailed analysis carried out to try and prove the diagnosis using known facts and other rules; backward chaining. This



procedure has the advantage of first indicating where to look for a solution rather than spending a lot of time retrieving unimportant data.

Once a fault is detected in the process under concern, the fault diagnosis system is called in order to advise the operator that a fault has occurred in a process component. The diagnosis task is performed through the comparison between the process variables real behaviour and the fault symptoms stored in the knowledge base. However, if the real behaviour of the process variables does not match any set of fault symptoms stored in the knowledge base, the fault diagnosis system is not able to identify the fault and an unsuccessful diagnosis occurs. At this stage a fault detection and diagnosis system with self-learning abilities will be desirable, which is considered in a further chapter.

Therefore, the fault diagnosis system implemented consists of a set of rules, a knowledge base with the fault symptoms, another knowledge base with the faults descriptions, and an inference engine. When the fault detection system observes a discrepancy between the real behaviour of the process variables and the predicted behaviour, the fault diagnosis system is triggered. At this stage, in order to locate a hypothetical fault, the fault diagnosis task starts.

The first linguistic value of the achieved process variable real behaviour, which is responsible for generating a discrepancy, is compared with the corresponding linguistic values of the fault symptoms, stored in the respective knowledge base. By this strategy, performed through forward chaining, a reduced set of fault candidates is generated. If the result of this procedure is a set of fault candidates that is empty, an unsuccessful diagnosis is achieved. However, if an empty set of fault candidates is not achieved, the fault diagnosis procedure will continue.

From the set of fault candidates, none of which is empty, the hypothetical diagnosis is performed by the “hypothesis-test” strategy. The procedure is first to generate a hypothesis from the set of fault candidates, then compare the real behaviour of the remaining sampled process variables with the fault symptoms. If they agree this hypothesis is retained. Otherwise, this procedure is continuously repeated until all fault candidates have been tested. Also here, if no hypothesis is retained, an unsuccessful diagnosis occurs.

To perform the diagnosis a recursive algorithm is implemented, which can work with any number of faults. Moreover, as the faults symptoms and the faults description are stored in a magnetic support we can consider a large malfunction hypothesis space that can be used for real time fault diagnosis.

The overall computational system described so far has been applied to the simulated mixing process described in chapter 4.. The next sub chapter presents the results achieved during the simulation studies.



## 6.6 Fault Detection and Diagnosis of the Mixing Process

The Mixing Process, shown in Figure 4.1 and described in chapter 4., has been used as a test bed of the computational system just introduced. Thus, the dynamic model, represented by equations (4.13) to (4.18), has been used to simulate the process under normal operation conditions, as well as under a fault situation. Control of level and temperature in tank 2 has been considered and, hence, the rule based controller, described in sub chapter 4.5, has been used. The fuzzy qualitative reasoning, which is used during the fault detection task, has been based on the qualitative model of the process. This model is a set of confluence's which are qualitative equations represented by expressions (4.19) to (4.26).

Therefore, since the detection of a fault is based on the comparison of the predicted behaviour with the real behaviour of process variables, when the qualitative simulator system is triggered the level and temperature of both tanks are predicted as described above, and the results passed to the fault detection system. Then, in the time instant  $t+1$ , the data fuzzification block samples the real behaviour of level and temperature in both tanks, and converts them into linguistic variables whose semantics are also represented by 4-tuple fuzzy numbers. The linguistic variables obtained by this way are then passed to the fault detection block, where they are compared with the fuzzy qualitative simulator results. According to the procedure described in sub chapter 6.4, a hypothetical fault is confirmed or denied. If a fault is confirmed then the fault diagnosis block is triggered in order to locate the faulty component. This task is performed according to the procedure described in sub chapter 6.5.

As presented in Figure 6.4, in the prototype implemented, the possible faults that may occur in the mixing process are listed in Table 6.5. All of them must be regarded as abrupt faults. Moreover, Table 6.5 also present the process variables used to simulate the faults. The qualitative

Faults	Representation
Hot water control valve fails low	$[\Delta Q_h] = \text{negative}$
Cold water control valve fails low	$[\Delta Q_c] = \text{negative}$
Hand valve 1 is blocked	$[\Delta Q_{o1}] = \text{negative}$
Hand valve 2 is blocked	$[\Delta Q_{o2}] = \text{negative}$
Hot water control valve fails high	$[\Delta Q_h] = \text{positive}$
Cold water control valve fails high	$[\Delta Q_c] = \text{positive}$

Table 6.5 - Single faults representation for “The Mixing Process”



values taken by the process variables under a failure situation are represented by the linguistic values "negative" and "positive", because the real fuzzy qualitative values taken from the descretized fuzzy quantity space, presented above, depend on the steady state conditions of the process when a single abrupt fault has been initiated. The behaviour of the process under a failure situation is achieved by changing the value of the corresponding variable in the dynamic model.

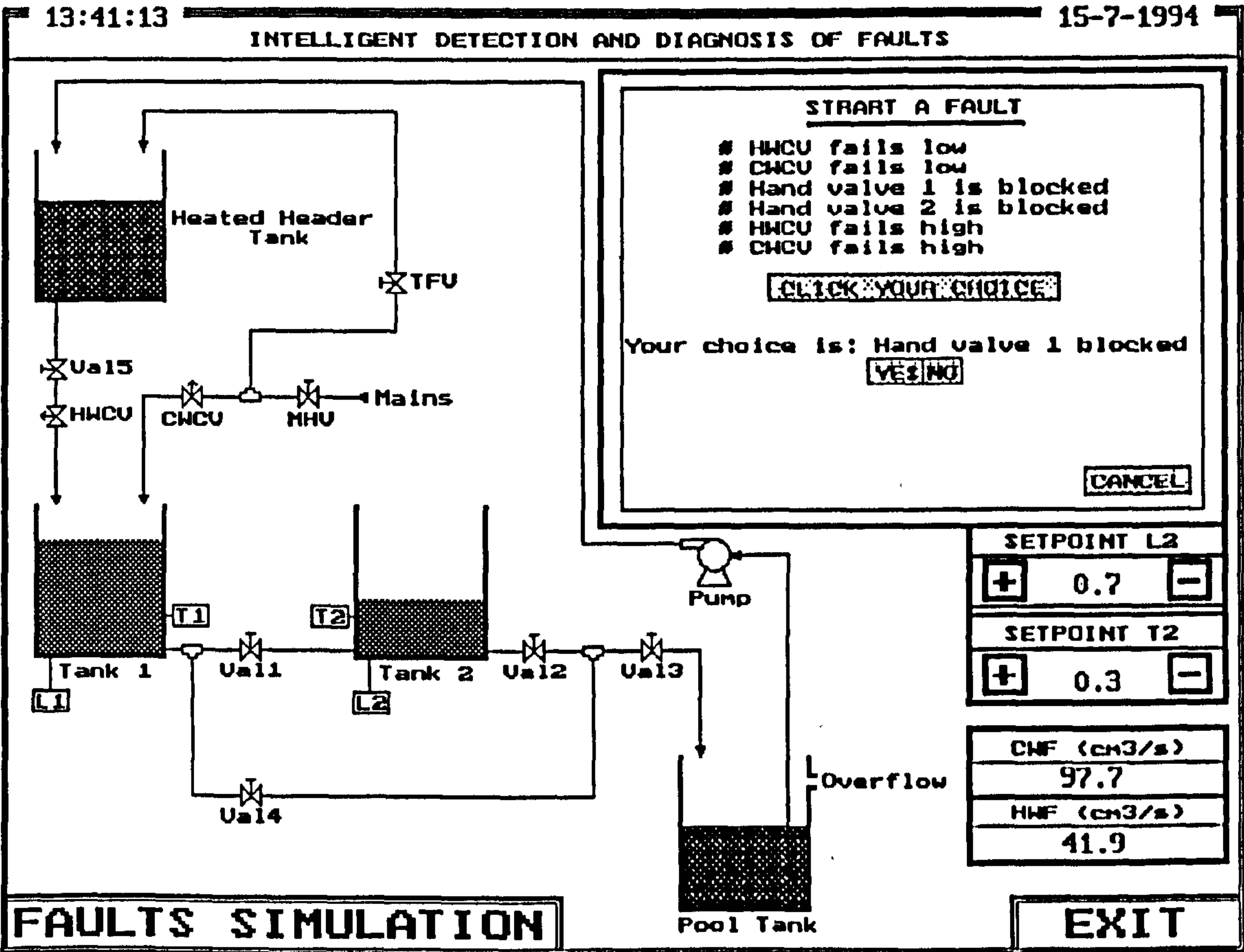


Figure 6.4 - Fault simulation

Moreover, as quoted in previous sub chapters, in order to save computational time and increase the system performance and reliability, an enable condition is defined, which should be satisfied in order to trigger the fault detection and diagnosis system. For the mixing process the definition of such enable conditions has been based on the process variables  $L_1$ ,  $L_2$ ,  $T_1$ ,  $T_2$ ,  $Q_h$  and  $Q_c$ . The notation used here is presented in chapter 4.. Furthermore, since the author has used a normalised range  $[-1, 1]$  to form the basis on which the fuzzy quantity space is descretized, for each variable a fuzzification parameter has been defined. Such fuzzification parameters are given in Table 6.6.

The fault detection system, based on the above procedure, has been applied successfully to the mixing process, where all the faults quoted in Table 6.5 have been detected in the first time that the qualitative simulator system is triggered; with the exception of the faults "Hand valve 2

Variables	Fuzzification parameters
$\Delta L_1$	1.0
$\Delta L_2$	1.1
$\Delta T_1$	0.75
$\Delta T_2$	0.7
$\Delta Q_h$	0.08
$\Delta Q_c$	0.095

Table 6.6 Fuzzification parameters for "The Mixing Process"

blocked" and "Hand valve 1 blocked", which are detected in the second time that the qualitative simulator has been triggered, after initialisation in some steady state conditions.

After the knowledge bases have been build with the fault symptoms and the corresponding fault descriptions, which have been acquired through simulation studies, the overall computational system has been successfully applied to the mixing process. During the experiments, all single faults mentioned in Table 6.5 were separately initiated with different steady state process

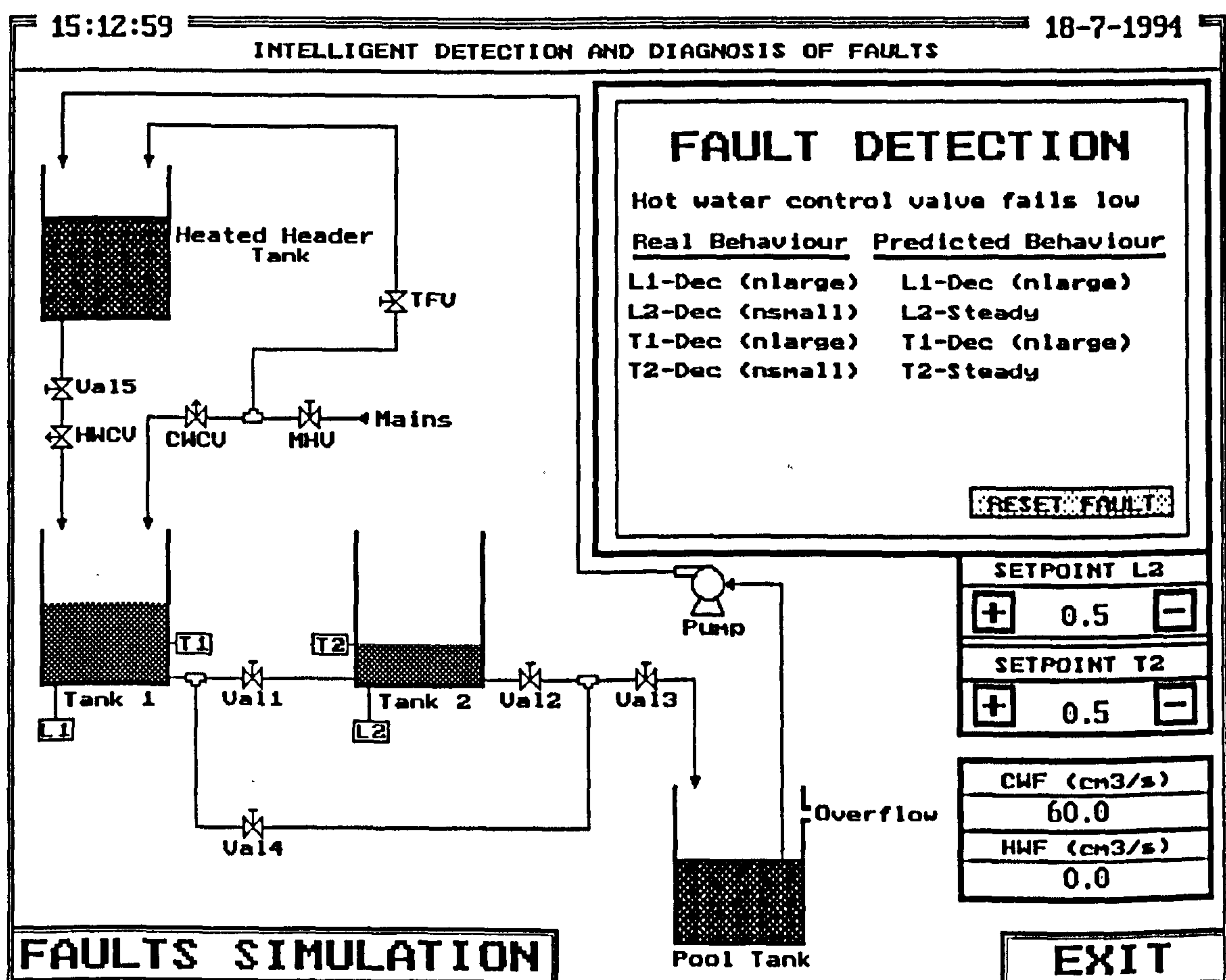


Figure 6.5 - Diagnostic of "Hot water control valve fails low" fault



conditions, and they were detected and diagnosed successfully. For instance, Figure 6.5 shows the diagnosis result achieved after fault, "Hot water control valve fails low", has been initiated with the setpoints of the controlled variables having the value 0.5.

## 6.7 Conclusions

This chapter has been concerned with applications of artificial intelligence techniques to on-line process control and fault diagnosis with the main emphasis on knowledge based systems for on-line process fault detection and diagnosis. Therefore, a computational system with a two layer configuration has been implemented through a TURBO C++ program. Process simulation and control of some process variables are performed in the lower layer. The on-line process control has been achieved through a rule based controller. Moreover, faults simulated in the lower layer could be detected and diagnosed through the fault detection and diagnosis system implemented in the upper layer. On-line fault detection and diagnosis is regarded as a supervisory task in the prototype developed.

A fault detection and diagnosis system based on fuzzy qualitative modelling is investigated. It is demonstrated that qualitative reasoning depends less on accurate process parameters and accurate measurement data and, therefore, a result obtained from qualitative reasoning is less accurate than that from quantitative reasoning. However, for the purpose of fault detection and diagnosis, accurate results are generally not needed and sometimes difficult to implement.

Ambiguity is a problem associated with qualitative reasoning. In order to solve this problem a fuzzy simulation algorithm has been developed. This algorithm allows a more detailed description of system variables than the classical qualitative simulation methods, through an arbitrary but finite discretisation of the quantity space. Moreover, the use of graded membership within a fuzzy quantity space, in a parameterized form, allows common sense knowledge to be incorporated in the basic description of the quantity space. The use of fuzzy relations and the fuzzy compositional rule of inference allows semi-quantitative information about the strength, as well as the sign, of functional relationships to be represented.

In order to improve efficiency and reliability in the diagnosis task an inference mechanism with forward and backward chaining abilities is used. First a set of fault candidates is generated through forward chaining and then, through backward chaining, the algorithm tries to confirm a hypothetical fault from the set of fault candidates. This procedure may be suitable for large scale processes where a big number of faults symptoms can be achieved, avoiding computational difficulties in performing real-time diagnosis.

The overall computational system has been successfully applied to a simulated mixing process. However, only single abrupt faults have been considered. An extension of the fault detection and diagnosis system implemented, which can cope with multiple simultaneous abrupt faults, is presented in the next chapter.



## Chapter 7

# On-Line Multiple Fault Diagnosis Through Fuzzy Qualitative Simulation

### 7.1 Introduction

Traditionally a process failure is diagnosed by skilled operators. The complexity of modern plants and the availability of inexpensive computer hardware allow us to develop automated fault detection and diagnosis systems. The broad requirements of an automated fault detection and diagnosis system are that it guarantee real time performance so that it can be useful in time critical situations, and that it possess an operator interface capable of displaying and updating results comprehensibly. Fundamental issues facing the design of a fault detection and diagnosis system include the kind of knowledge to represent, the representation scheme and the inference strategy that performs the actual diagnosis.

Based on artificial intelligence techniques a huge number of fault diagnosis systems have been pointed out by several researchers. However, a modest amount of work has been done on the use of automated systems to detect and diagnose multi-simultaneous faults on the process under concern. In the single fault case, only one physical component is the source of plant failure, whereas in the multiple fault case, more than one physical component is the source. Narayanan and Viswanadham (1987), Padalkar *et al.* (1991) and Rao *et al.* (1987) diagnose single faults, and offer some assistance in localising multiple faults. De Kleer and Williams (1987) used a process of backwards and forwards propagation to accept or rule out single faults, and then successively investigate more complex multiple fault cases. Guan and Graham (1994) used a digraph approach to detect and diagnose single and multiple faults. Watanabe *et al.* (1994) used a hierarchical

artificial neural networks technique to diagnose multiple simultaneous faults. The author presents a different approach to detect and diagnose multiple simultaneous faults based on a fuzzy qualitative simulation algorithm.

The difficulty with diagnosing multiple faults based on the classical mathematical models or state space models is that the process model needs to be almost perfect and extensive calculations are needed. If there are errors in the model, they may manifest as faults, thus yielding false alarms. As in the approach presented in the last chapter, there is a motivation in this context, to avoid effort and expense of creating, maintaining and computing with rigorous dynamic mathematical models of large-scale processes, by focusing on qualitative indicators of process condition (Oyeleye and Kramer 1988).

Studies of human diagnostic strategies show that a hypothesis/test procedure is frequently used by operators to mentally simulate the effect of the hypothesised malfunction on process behaviour (Rasmussen 1980). However if the malfunction is a consequence of more than one component failure the operators may have difficulty to determine and identity the process faults. Because time constraints are critical, hesitation as well as inappropriate action could lead to disaster.

In previous research work, which is described in chapter 6., the author developed a hierarchical computational system based on qualitative simulation to detect and diagnose single faults introduced in a pilot mixing process plant simulation. In the current chapter an extension of the system developed to cope with the detection and diagnosis of multiple simultaneous faults is presented. The goal is to develop a real time fault detection and diagnosis system able to accurately diagnose both cases.

A set of double simultaneous faults corresponding to an AND set in the single fault space is considered as already pointed out by Watanabe *et al.* (1994). So, in general, if the single fault space is represented by,

$$\{F_1, F_2, F_3, \dots, F_n\} \quad (7.1)$$

the AND set for double simultaneous faults, must be represented by,

$$\{(F_1, F_2), (F_1, F_3), \dots, (F_1, F_n), \dots, (F_2, F_3), \dots, (F_2, F_n), \dots, (F_{n-1}, F_n)\} \quad (7.2)$$

However, while the authors quoted used a hierarchical artificial neural networks approach, the diagnosis system presented here is based on a fuzzy qualitative simulation technique. Therefore, the current approach is based on deep knowledge of the process under consideration.



The view of qualitative process behaviour advanced by the author is that the effects of a fault and/or faults are propagated from one variable to another in the form of linguistic variables whose semantics are represented by fuzzy numbers, ultimately satisfying steady-state process constraints. As quoted in the previous chapter, these constraints are sets of algebraic equations, represented qualitatively as steady-state confluence equations, which are derived from the dynamic model of the process under consideration according to the proposal pointed out by de Kleer and Brown (1984).

The knowledge corresponding to the single and multiple fault symptoms is stored in a knowledge base. However the use of fuzzy sets to describe the process variables real behaviour lead to a more than one fault symptom by each fault considered, each one corresponding to different steady state conditions of the process under concern. So in order to avoid computational difficulties in performing a real time diagnosis, the inference engine with forward and backward chaining abilities, described in the last chapter, has been used. This procedure has been shown to be very efficient even when the number of faults increases significantly.

In the remainder of this chapter, a description of the fault detection and diagnosis technique used is presented as well as a discussion of the problems found out with the extension of the single fault detection and diagnosis system to detect and diagnose multiple simultaneous faults. Sub chapter 7.2 presents the architecture of this new approach. In sub chapter 7.3 the fault diagnosis task under the new architecture of the fault detection and diagnosis system is discussed. In sub chapters 7.4 and 7.5 the results achieved with the application of the system to detect and diagnose double simultaneous faults introduced respectively in simulation studies of a mixing process and of a continuous stirred tank reactor are presented. Sub chapter 7.6 provides some concluding remarks.

## 7.2 System Architecture

The system architecture of the fault detection and diagnosis system, which has single and double simultaneous fault detection and diagnosis abilities, is depicted in Figure 7.1. As can be seen the configuration of the system is similar to the architecture of the system described in the last chapter, which has only single abrupt fault detection and diagnosis capabilities. The computational system implemented still has a hierarchical structure of two layers configuration. In the lower layer a process is simulated through it's dynamic model in a differential equations form and to solve the ordinary differential equations a fifth-order Runge-Kutta method is still used. It is also in this layer that a fault or double simultaneous faults can be initiated in order to achieve the process behaviour

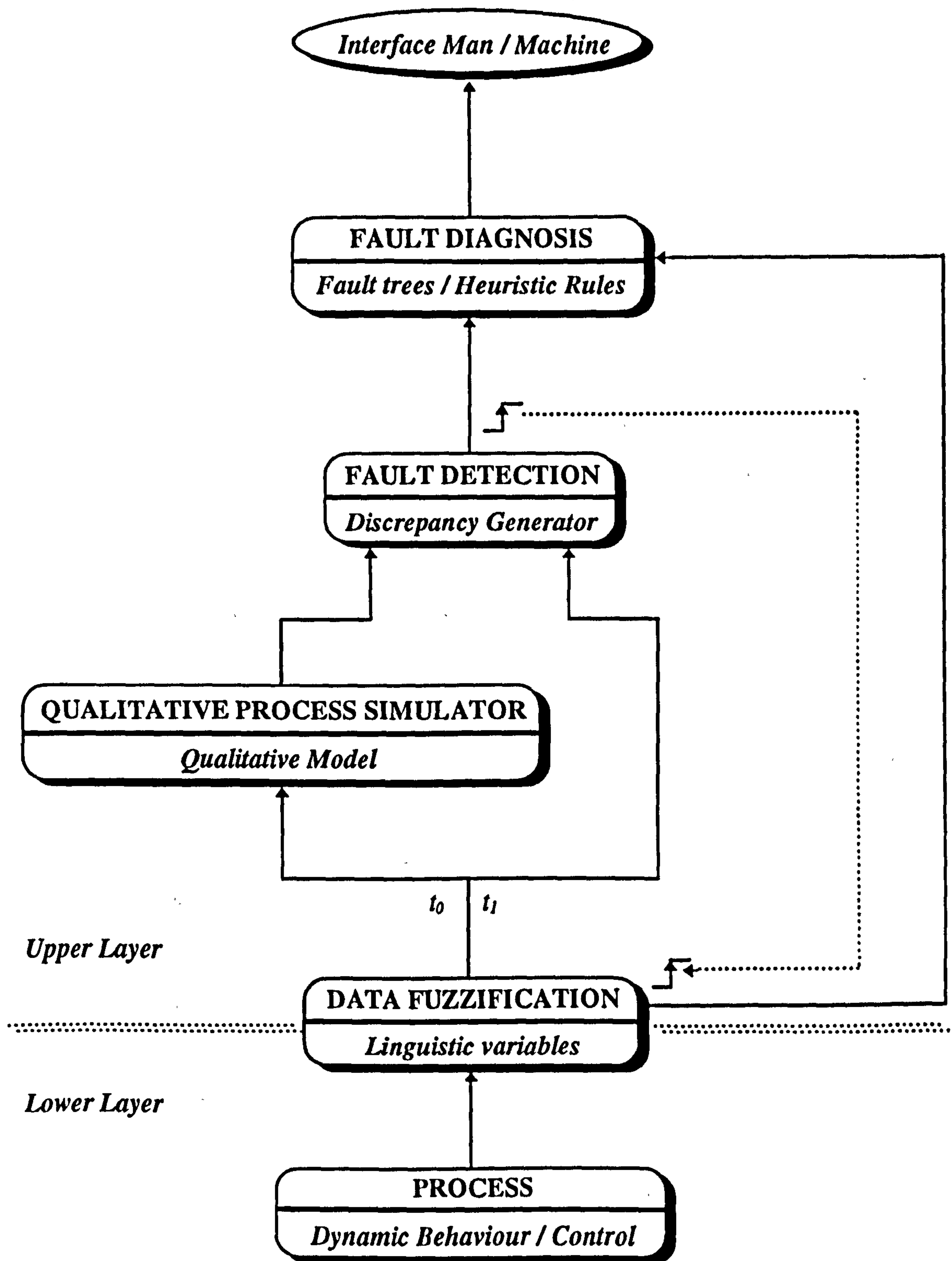


Figure 7.1 - General scheme of a computer assisted multi-faults diagnosis system

under a failure situation. The present approach has been applied during simulation studies conducted with the mixing process described in chapter 4., as well as during simulation studies performed with the continuous stirred tank reactor (CSTR) described in chapter 5.. Therefore, the process variables control for the first case has been achieved through a rule based controller while for the second example of an industrial process classic controllers, such as PID, PI and P, have been used.



As can be seen from Figure 7.1, the interface between the two layers is still performed by a Data Fuzzification block. The rates of change of the process variables behaviour are sampled and then converted into linguistic variables whose semantics are represented by fuzzy numbers. However, during the current research work, it has been observed that the adoption of fuzzy subsets has a direct advantage over the traditional crisp representation when considering granularity.

So, let us to define the property granularity. If  $x_1, x_2 \in X$  characterise similar things or stand for similar properties of a variable  $X$ , then the relevant qualitative values of  $x_1$  and  $x_2$  are equal to each other.

In fact, if we intend to describe the qualitative values of the process variables rate of change only in terms of the crisp subsets of the underlying real range, the mapping from the real range to a quantity space will result in the search for the limits of the real numbers at the boundaries among adjacent qualitative values within the quantity space. This usually incurs severe difficulties in determining these limits (Shen and Leitch 1993).

The fuzzy representation of qualitative values is indeed more general than ordinary interval representations, since it can represent not only the information stated by a well defined real interval but the knowledge embedded in the soft boundaries of the interval is also represented. Thus, fuzzy quantity space reduces the boundary interpretation problem, which is achieved through the description of a gradual rather than an abrupt change in the degree of membership of which a physical quantity is mapped into a particular qualitative value.

Since it has been observed that the parametric representation of the membership function of a fuzzy number is a good approximation of the result obtained from using the extension principle to evaluate arithmetic functions with fuzzy numbers, and has a more limited computational overhead, in the present approach the author has still used the 4-tuple fuzzy numbers as semantics of the qualitative values taken by the process variables. Thus, the set of qualitative linguistic values used in this system remain the same as used in the approach described in the last chapter.

However, in the present approach the supervisory system, which has been implemented in the upper layer, has the abilities to detect and diagnose single and double simultaneous abrupt faults simulated in the lower layer. Both the multiple simultaneous fault detection and diagnosis tasks and the single faults detection and diagnosis tasks are performed in a similar way. The procedures for single faults have been described in chapter 6.. Hence, in this chapter only a brief description of the techniques is presented. However, with the extension of the system to cope with multiple simultaneous faults, some modifications have been performed and are discussed in the remainder of this chapter.

It is worth noting that, in order to save computational time and increase system's reliability, an enable condition, which consists of several constraints on the sampled process



variables rate of change, is defined. Then, only when this enable condition is satisfied, is the upper layer triggered in order to detect and diagnose a hypothetical fault or faults in the process under concern.

A man/machine interface is used to keep the operator informed about any fault occurred in the process under consideration. The location of a fault or faults in the process will be given through this interface, as well as the reasoning followed during the fault detection task. Moreover, during the simulation studies conducted with both processes quoted above, this interface has been used to simulate faults in the process. Thus, the interaction between the computational system and the operator in order to start a fault or faults simulation occurs into levels. At the first level the operator must choose between single faults or multiple simultaneous faults as Figure 7.2 shows. When the operator chooses single or multiple faults, the second level of the man/machine interface allows the operator to introduce a single or a double simultaneous fault respectively, in the process under consideration. Figure 7.3 shows an example of the second level's interface implemented to the continuous stirred tank reactor.

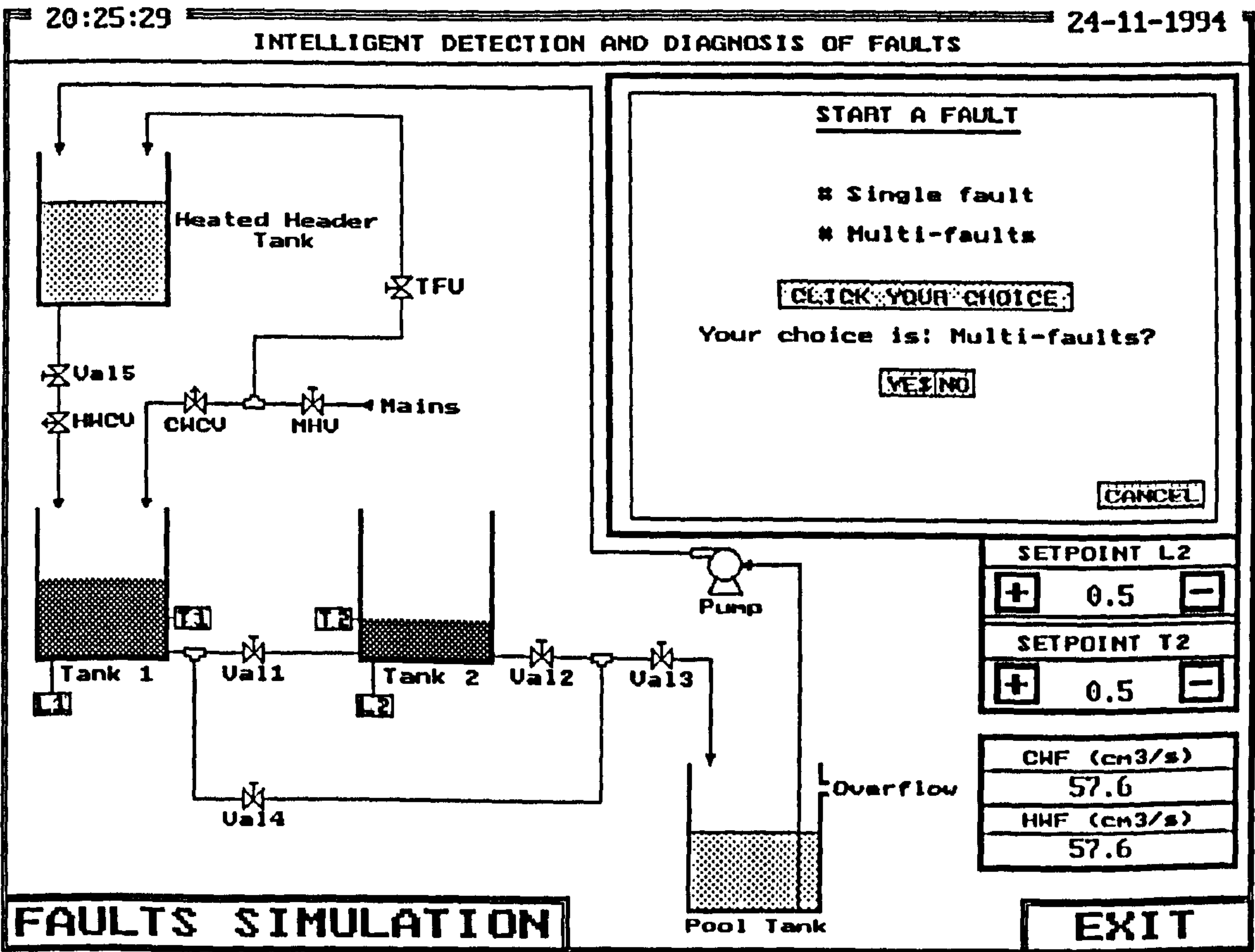


Figure 7.2 - First level of the interface man/machine in order to start a fault simulation situation

The mixing process simulation has been achieved through its dynamic model represented by equations (4.13) to (4.18), while the qualitative reasoning performed during the fault detection



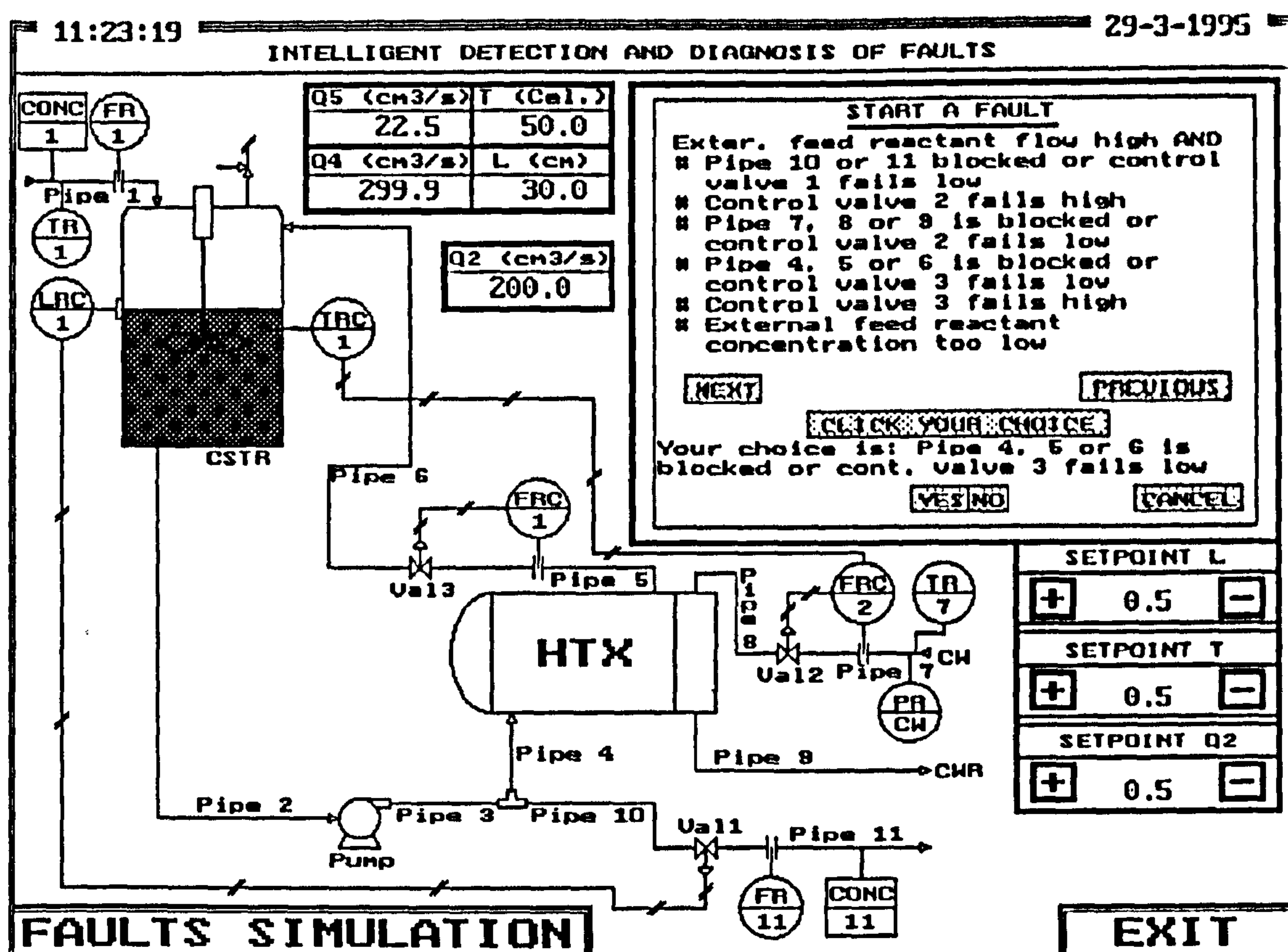


Figure 7.3 - Interface man/machine to start the simulation of double simultaneous faults in "The Continuous Stirred Tank Reactor"

task, has been achieved through the qualitative model in a form of set of confluence's represented by equations (4.19) to (4.26). The simulation of the CSTR process has been obtained using the dynamic model represented by equations (5.24) to (5.31) and the qualitative simulation of this process has been performed through the qualitative model represented by equations (5.55) to (5.58).

As quoted above, the fault detection and diagnosis system presented in this chapter, which has the ability to detect and diagnose single and multiple simultaneous abrupt faults, is an extension of the system described in chapter 6.. Thus, to cope with multiple simultaneous faults, a main change has been performed in the fault diagnosis block, represented in Figure 7.1. Therefore, the fault diagnosis reasoning, which has been followed under the current approach, is described in sub chapter 7.3.

## 7.3 Fault Diagnosis Reasoning

A knowledge based system has been used for fault diagnosis purposes. In the present approach two knowledge bases are used, as shown in Figure 7.4. In the first one, the fault

symptoms are stored, while in the second one the corresponding fault descriptions, given by the operators, are saved. An inference engine with forward and backward chaining abilities, has been implemented. Through forward chaining, a general analysis is carried out to determine what faults are possible and to generate a reduced set of fault candidates. Through backward chaining, a diagnosis is then selected and a more detailed analysis carried out to try and prove the diagnosis using known facts and other rules. This procedure has the advantage of first indicating where to look for a solution rather than spending a lot of time retrieving unimportant data. As the results presented in the next two sub chapters show, it has been observed that, following this strategy, the inference engine performs very satisfactorily, even when the number of fault symptoms increase. Such results have been obtained during simulation studies conducted with both processes quoted above.

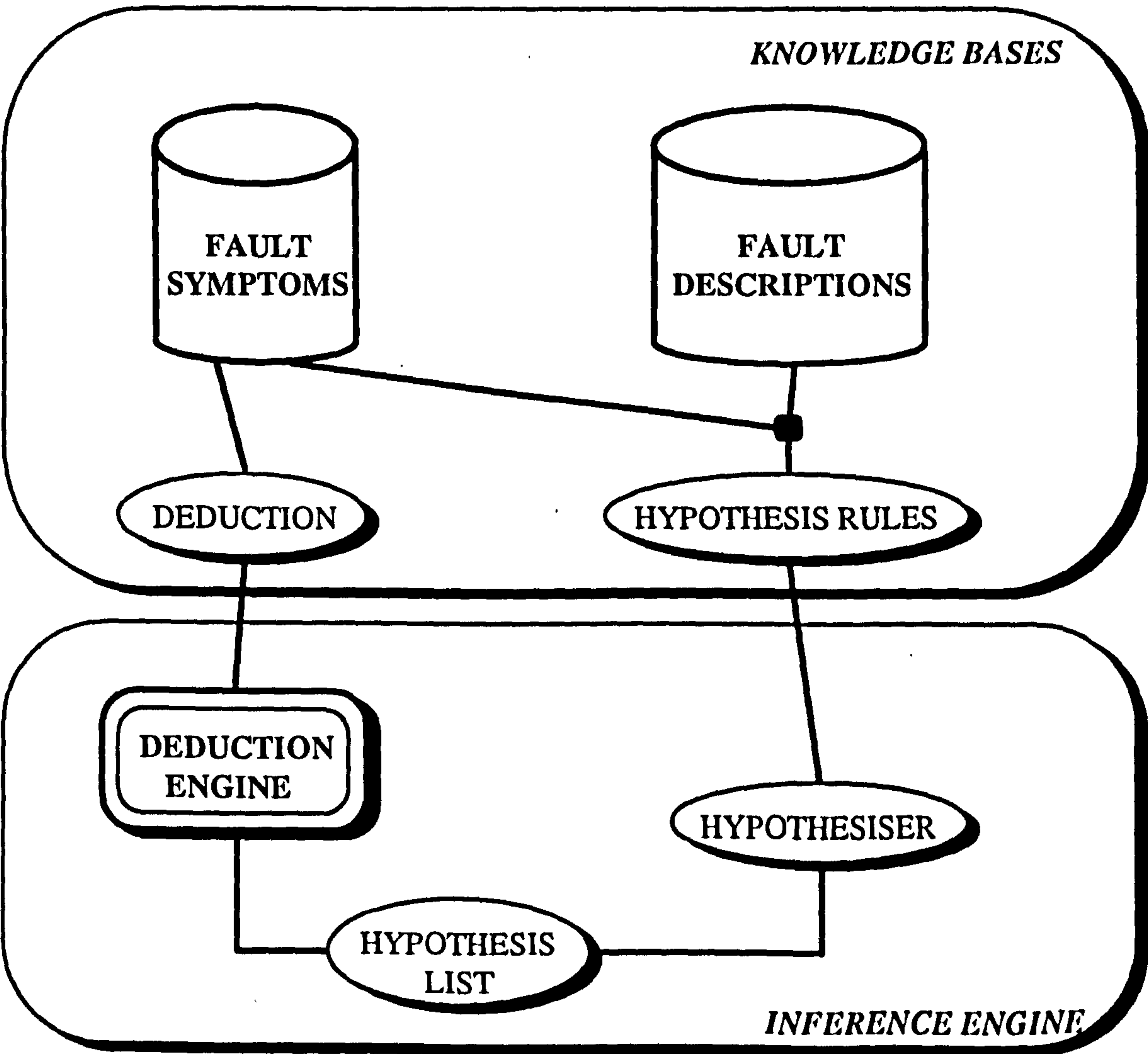


Figure 7.4 - Fault diagnosis system organisation

As in the fault diagnosis system described in the last chapter, the fault diagnosis procedure is performed through the comparison between the process variables real behaviour and the fault



symptoms. However, with the system extension to cope with multi-simultaneous faults some changes have been done, which are discussed in the remainder of this sub chapter.

Since each different fault takes a different time to affect the process behaviour, with the system architecture used to diagnose single faults, under a multi-simultaneous faults situation, only the fault with a quick effect was diagnosed. It has been observed that by using the process variables rate of change, corresponding to the time instant when the fault detection and diagnosis system is triggered, it is possible to overcome this problem and to increase the reliability of the system to cope with multi-simultaneous faults. Then, the main change in the fault diagnosis reasoning was not to limit consideration to the process variables behaviour in the time instant  $t_i$  after a fault has been detected, as in the single fault detection and diagnosis approach. Together with these process variables, the behaviour of other process variables, which must be directly affected by certain faults considered, in the instant of time that the supervisor system is triggered, are also considered. The rationale for this is that, for the process variables directly affected by certain faults, no changes in their values in the time instant  $t_i$  will be observed after such abrupt faults have been detected. Therefore, if we consider the changes in the values of these process variables in the time instant  $t_i$ , the fault detection input space will be reduced and the problem quoted above arises again.

So, when the fault detection system observes a discrepancy between the process variables real behaviour and the predicted one, the fault diagnosis task starts. The linguistic values of the process variables real behaviour used by the Fault Detection block are passed to the Fault Diagnosis block, as well as the linguistic values of process variables considered as directly affected by faults, corresponding to the time instant that the fault detection block was fired. This set of linguistic values is retained in the Data Fuzzification block until the fault diagnosis system has been triggered.

Once the fault diagnosis system has been fired, the first linguistic value, which is received from the fault detection system and was responsible for generating a discrepancy, is compared with the corresponding linguistic values of the fault symptoms stored in the respective knowledge base. As described above, following this strategy, performed through forward chaining, a reduced set of fault candidates is generated. From the set of fault candidates, the hypothetical diagnosis is achieved through backward chaining, following a “hypothesis-test” procedure. The procedure is first to generate a hypothesis from a set of fault candidates and, then to compare the real behaviour of the remaining variables with the fault symptoms. If they agree this hypothesis is retained. Otherwise, this procedure is continuously repeated until all fault candidates have been tested. However, if no hypothesis is retained, an unsuccessful diagnosis occurs and a self-learning system will be desirable to avoid such situations. Hence, in the next chapter a self-learning system for the present approach is described.



Successful results achieved during simulation studies conducted with the mixing process, and with the CSTR, are presented in the next two sub chapters, respectively.

## 7.4 Fault Detection and Diagnosis of the Mixing Process

In previous research work, single abrupt faults in the mixing process have been successfully detected and diagnosed by a fault detection and diagnosis system described in the last chapter. Here, results achieved through the approach described in this chapter are presented. The aim is to detect and diagnose double simultaneous faults initiated in the mixing process.

The set of double simultaneous faults considered in the present approach is derived from the single faults space through an AND operation. So, as in previous research work, with the single faults space considered as represented in Table 6.5, applied to expression (7.2), the set for double simultaneous faults, presented in Table 7.1, is achieved.

To build up the knowledge base with the fault symptoms, simulation studies have been conducted with the mixing process, where all the single and double simultaneous abrupt faults, quoted above, have been initiated with all the possible setpoints combinations of the controlled variables level in tank 2,  $L_2$ , and temperature in tank 2,  $T_2$ , taking values between 0.1 and 1.0 with a stepsize of 0.1. Thus, as the process variables behaviour is described by fuzzy numbers, as stated above, we can have more than one fault symptom for each fault according to the different steady state conditions of the process. Hence, the result achieved during a fault symptoms learning phase produced a knowledge base with 278 fault symptoms. Even with this number of fault symptoms, all the single and double simultaneous faults, quoted above, have been detected and diagnosed successfully with different steady state conditions. It has been observed that the inference engine, with forward and backward chaining abilities, performed very efficiently providing the diagnosis in less than 1 second on a 386 PC (25 MHz) fitted with a mathematical co-processor.

For example, Figure 7.5 shows the diagnostic achieved after the double simultaneous faults, “Hot water control valve fails low AND Hand valve 1 is blocked”, have been initiated with the setpoint of the controlled variable level in tank 2 having the value 0.6, and the setpoint of the other controlled variable, temperature in tank 2, having the value 0.4. Figure 7.6 shows the disturbance in the controlled variables, as well as the Rule Based Controller actions in order to correct the disturbance effect after the faults have been reset.

As is described in sub chapter 7.3, with the extension of the fault detection and diagnosis system to cope with multi-simultaneous abrupt faults some modifications have been done to the system architecture. In the previous research work the diagnosis reasoning of single faults was



Faults	Representation
Hot water control valve fails low AND Cold water control valve fails low	$[\Delta Q_h] = \text{negative AND}$ $[\Delta Q_c] = \text{negative}$
Hot water control valve fails low AND Hand valve 1 is blocked	$[\Delta Q_h] = \text{negative AND}$ $[\Delta Q_{o1}] = \text{negative}$
Hot water control valve fails low AND Hand valve 2 is blocked	$[\Delta Q_h] = \text{negative AND}$ $[\Delta Q_{o2}] = \text{negative}$
Hot water control valve fails low AND Cold water control valve fails high	$[\Delta Q_h] = \text{negative AND}$ $[\Delta Q_c] = \text{positive}$
Cold water control valve fails low AND Hand valve 1 is blocked	$[\Delta Q_c] = \text{negative AND}$ $[\Delta Q_{o1}] = \text{negative}$
Cold water control valve fails low AND Hand valve 2 is blocked	$[\Delta Q_c] = \text{negative AND}$ $[\Delta Q_{o2}] = \text{negative}$
Cold water control valve fails low AND Hot water control valve fails high	$[\Delta Q_c] = \text{negative AND}$ $[\Delta Q_h] = \text{positive}$
Hand valve 1 is blocked AND Hand valve 2 is blocked	$[\Delta Q_{o1}] = \text{negative AND}$ $[\Delta Q_{o2}] = \text{negative}$
Hand valve 1 is blocked AND Hot water control valve fails high	$[\Delta Q_{o1}] = \text{negative AND}$ $[\Delta Q_h] = \text{positive}$
Hand valve 1 is blocked AND Cold water control valve fails high	$[\Delta Q_{o1}] = \text{negative AND}$ $[\Delta Q_c] = \text{positive}$
Hand valve 2 is blocked AND Hot water control valve fails high	$[\Delta Q_{o2}] = \text{negative AND}$ $[\Delta Q_h] = \text{positive}$
Hand valve 2 is blocked AND Cold water control valve fails high	$[\Delta Q_{o2}] = \text{negative AND}$ $[\Delta Q_c] = \text{positive}$
Hot water control valve fails high AND Cold water control valve fails high	$[\Delta Q_h] = \text{positive AND}$ $[\Delta Q_c] = \text{positive}$

Table 7.1 - Double simultaneous faults representation for “The Mixing Process”

based on the behaviour of the process variables level and temperature in tanks 1 and 2, respectively,  $L_1$ ,  $L_2$ ,  $T_1$  and  $T_2$ , at the time instant,  $t_I$ , after the fault detection and diagnosis system has been triggered. In the present approach, in order to be able to detect and diagnose multi-

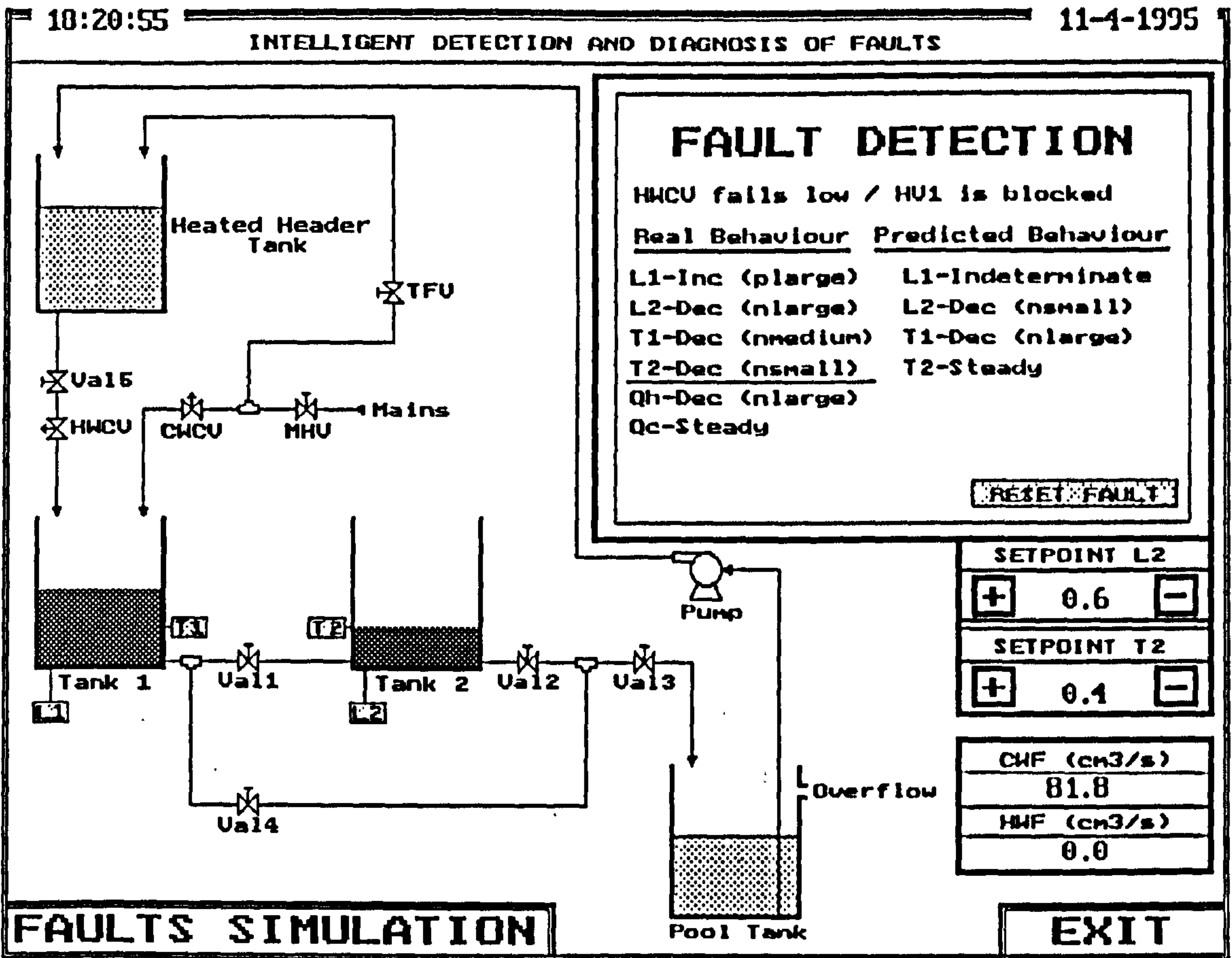


Figure 7.5 - Diagnostic of "Hot water control valve fails low AND Hand valve 1 is blocked" faults

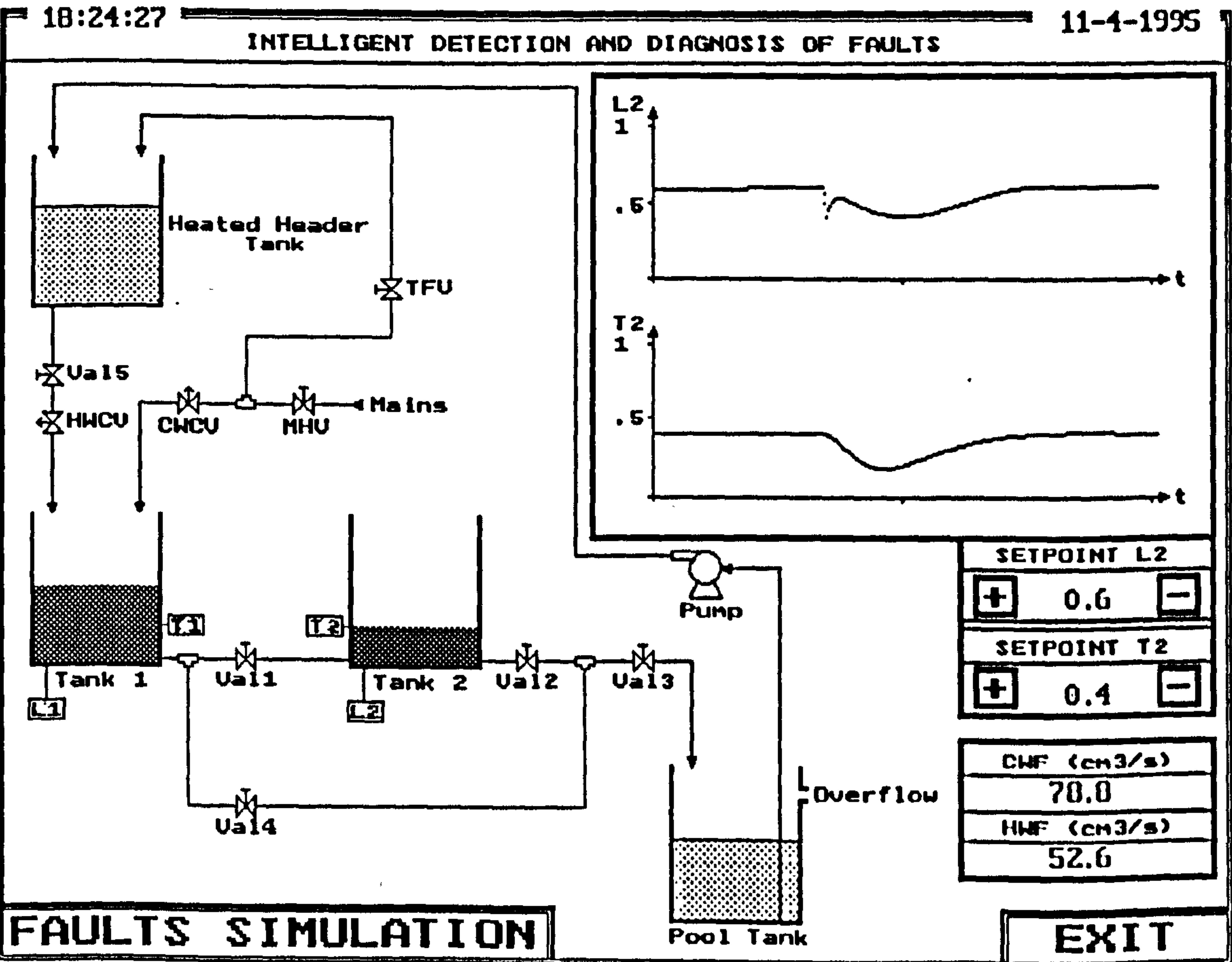


Figure 7.6 - Disturbance effect after faults, "Hot water control valve fails low AND Hand valve 1 is blocked", have been occurred



simultaneous faults, the diagnosis reasoning is based on the behaviour of the process variables quoted above, as well as on the behaviour of the process variables hot water input flow and cold water input flow, respectively,  $Q_h$  and  $Q_c$  in the time instant corresponding to the fault detection and diagnosis system trigger, as shown in Figure 7.5. Otherwise, as we are considering abrupt faults and each fault has different time effects in the process behaviour, only the fault with a quick effect will be detected. The procedure is to perform the diagnosis based on the behaviour of the process variables used in the single fault detection and diagnosis approach and on the behaviour of the process variables  $Q_h$  and  $Q_c$ , which are directly affected by the hot water control valve faults and cold water control valve faults respectively, as described in the previous sub chapter.

The diagnosis reasoning of double simultaneous faults follow a similar procedure as described in chapter 6. for the single abrupt faults. The process variables behaviour is compared with a set of fault symptoms stored in a knowledge base, in order to identify the fault symptoms that match the real process variables behaviour. Following this procedure single or double simultaneous faults are located in the process under concern in a similar way.

In order to demonstrate the robustness of the proposed fault detection and diagnosis system, a more complicated process has been used in order to test the approach. As for the mixing process, single and double simultaneous abrupt faults have been considered. Therefore, simulation studies have been conducted with a continuous stirred tank reactor and the results achieved, are presented in the next sub chapter.

## **7.5 Fault Detection and Diagnosis of the Continuous Stirred Tank Reactor**

To test the computational system implemented, another example of an industrial process has been taken, which is the continuous stirred tank reactor described in chapter 5.. The on-line fault detection and diagnosis system for the continuous stirred tank reactor is similar to that of the mixing process described in the last sub chapter. The process behaviour under a failure situation is achieved by changing the process variable values in the dynamic model, according to the fault chosen by the operator. The single faults considered in this process to test the present fault detection and diagnosis system are listed in Table 7.2, as well as the corresponding variables used to initiate the fault.

Each variable has its own underlying numeric range of values. For simplicity, however, we still use a normalised range  $[-1, 1]$  to form the basis on which the fuzzy quantity space is



discretized. Hence, the fuzzy quantity space presented in sub chapter 6.2 was adopted, with the qualitative values represented by seven 4-tuple parametric fuzzy numbers as shown in Figure 6.2. However, as can be seen in Table 7.2, the qualitative values taken by the process variables rate of change, under a failure situation, are represented by the qualitative values “negative” and “positive”. This is because the real qualitative values taken from the fuzzy quantity space quoted depends on the steady state conditions of the process in the time instant that the operator initiated the fault.

Faults	Representation
Pipe 1 is blocked	$[\Delta Q_1] = \text{negative}$
External feed reactant flow high	$[\Delta Q_1] = \text{positive}$
Pipe 2 or 3 blocked or pump fails	$[\Delta(\Delta P)] = \text{negative}$
External feed reactant temperature high	$[\Delta T_1] = \text{positive}$
External feed reactant temperature low	$[\Delta T_1] = \text{negative}$
Pipe 10 or 11 is blocked or control vale 1 fails low	$[\Delta A_4] = \text{negative}$
Control valve 2 fails high	$[\Delta A_5] = \text{positive}$
Pipe 7, 8 or 9 is blocked or control valve 2 fails low	$[\Delta A_5] = \text{negative}$
Control valve 1 fails high	$[\Delta A_4] = \text{positive}$
Pipe 4, 5 or 6 is blocked or control valve 3 fails low	$[\Delta A_2] = \text{negative}$
Control valve 3 fails high	$[\Delta A_2] = \text{positive}$
External feed reactant concentration too low	$[\Delta C_{a0}] = \text{negative}$

Table 7.2 - Single faults representation for the "Continuous Stirred Tank Reactor"

The fault detection and diagnosis system is triggered only when a pre-defined enable condition, which consists of several constraints on the sampled process variables, is satisfied. For the CSTR process the sampled process variables used to defined that enable condition are  $L$ ,  $C_a$ ,  $C_b$ ,  $T$ ,  $Q_1$ ,  $T_1$ ,  $C_{a0}$ ,  $Q_2$ ,  $Q_5$ ,  $T_5$  and  $Q_4$ , which are measurement process variables. The notation used in this sub chapter for measurement and controlled variables of the continuous stirred tank reactor is defined in chapter 5..

Moreover, the constraints are defined in a form of linguistic values. This means that to satisfy the enable condition the sampled process variables values must be converted into linguistic values, which is performed by the Data Fuzzification block. Therefore, the fault detection and diagnosis system will be triggered if at least one of the process variables quoted above takes a linguistic value greater than “*psmall*” or less than “*nsmall*”. However, as we use a normalised



range  $[-1, 1]$  to form the basis on which the fuzzy quantity space is discretized, a fuzzification parameter must be defined for each variable. Table 7.3 presents the fuzzification parameters adopted in the current approach, which have been adjusted following a similar procedure whose details have been described in chapter 6..

Variables	Fuzzification values
$L$	0.23
$C_a$	32.0
$C_b$	85.0
$T$	0.35
$T_2$	0.2
$\Delta P$	0.005
$P$	0.2
$Q_1$	0.003
$T_1$	0.25
$C_{a0}$	1.0
$Q_2$	0.0075
$Q_4$	0.0013
$Q_5$	0.043
$T_5$	0.25
$A_2$	1.0
$A_4$	0.33

Table 7.3 - Fuzzification values of the CSTR process variables

Once the fault detection and diagnosis system has been fired the real values of the process variables rate of change of  $L$ ,  $T$ ,  $C_a$ ,  $C_b$ ,  $T_1$ ,  $T_2$ ,  $Q_1$ ,  $Q_2$ ,  $\Delta P$  and  $C_{a0}$ , and the real value of the controlled variable rate of change of  $A_4$ , are passed to the Qualitative Process Simulator block in the form of linguistic values, whose semantics are represented by 4-tuple fuzzy numbers as quoted above. In this block, they are propagated through the qualitative model of the process under consideration, which is a set of confluence's represented by equations (5.55) to (5.58). The aim is to predict the behaviour of the sampled process variables  $L$ ,  $C_a$ ,  $C_b$  and  $T$ , in the time instant  $t_1$ , after the fault detection and diagnosis system has been triggered. Hence, the rates of change, of such process variables obtained by this way, are qualitative values represented by fuzzy numbers expressed in a parametric form.



Then, the predicted linguistic values by the qualitative process simulator are passed to the Fault Detection block where they are compared with the real values of the process variables rate of change in the time instant  $t_l$ . The result of this comparison may, or may not, be a discrepancy between the predicted behaviour and the real one. If a discrepancy is generated, then it is considered that a fault has occurred in the CSTR process and, in order to obtain a location of this fault, the Fault Diagnosis block will be fired. If no discrepancy is achieved, then the hypothetical failure situation will be denied.

Once a fault has been detected, the Fault Diagnosis block is triggered in order to determine the fault location in the process. The goal is to discover which process component is faulty, in order to give such information to the operator through the interface man/machine. This task is performed by comparing the real behaviour of the process variables rate of change, in a linguistic form, with the fault symptoms stored in the knowledge base. In order to avoid computational difficulties to achieve a real time diagnosis, the fault diagnosis task has been performed by using the inference engine with forward and backward chaining abilities, which is described above. Good performance and reliability of this procedure has been observed.

During simulation studies and experiments, it has been found that the diagnosis task is well performed for single faults, only with the knowledge corresponding to the process variables  $L$ ,  $C_a$ ,  $C_b$  and  $T$ , in the time instant  $t_l$ . However, as quoted above for the mixing process, when multiple simultaneous faults are considered it has been observed that more information is needed in order to perform a reliable diagnosis. Otherwise, since each different fault takes a different time to affect the process behaviour, only the fault with a quick effect will be diagnosed. Then, in order to overcome this problem in the CSTR process, the diagnosis task is performed by using the knowledge quoted above together with the real process variables rates of change of  $T_l$ ,  $Q_1$ ,  $Q_2$ ,  $Q_4$  and  $Q_5$ , in the time instant  $t_0$ . The aim is to consider some process variables which are directly affected by the failure component, as described in the previous sub chapter.

After a phase of simulation studies performed to acquire the fault symptoms of single abrupt faults, the fault detection and diagnosis system has been successfully applied to the continuous stirred tank reactor. During the experiments, all single faults mentioned above were separately initiated with different steady state conditions of the process, and were detected and diagnosed successfully. For instance, Figure 7.7 shows the diagnosis result after the fault, "Pipe 1 is blocked", has been initiated with all the setpoints of controlled variables, respectively  $L$ ,  $T$  and  $Q_2$ , having the value 0.5.

As for the mixing process, the multiple simultaneous abrupt faults considered in the current approach are double faults initiated simultaneously in the continuous stirred tank reactor. Hence, according to expression (7.2), the set of double simultaneous faults is achieved through an AND



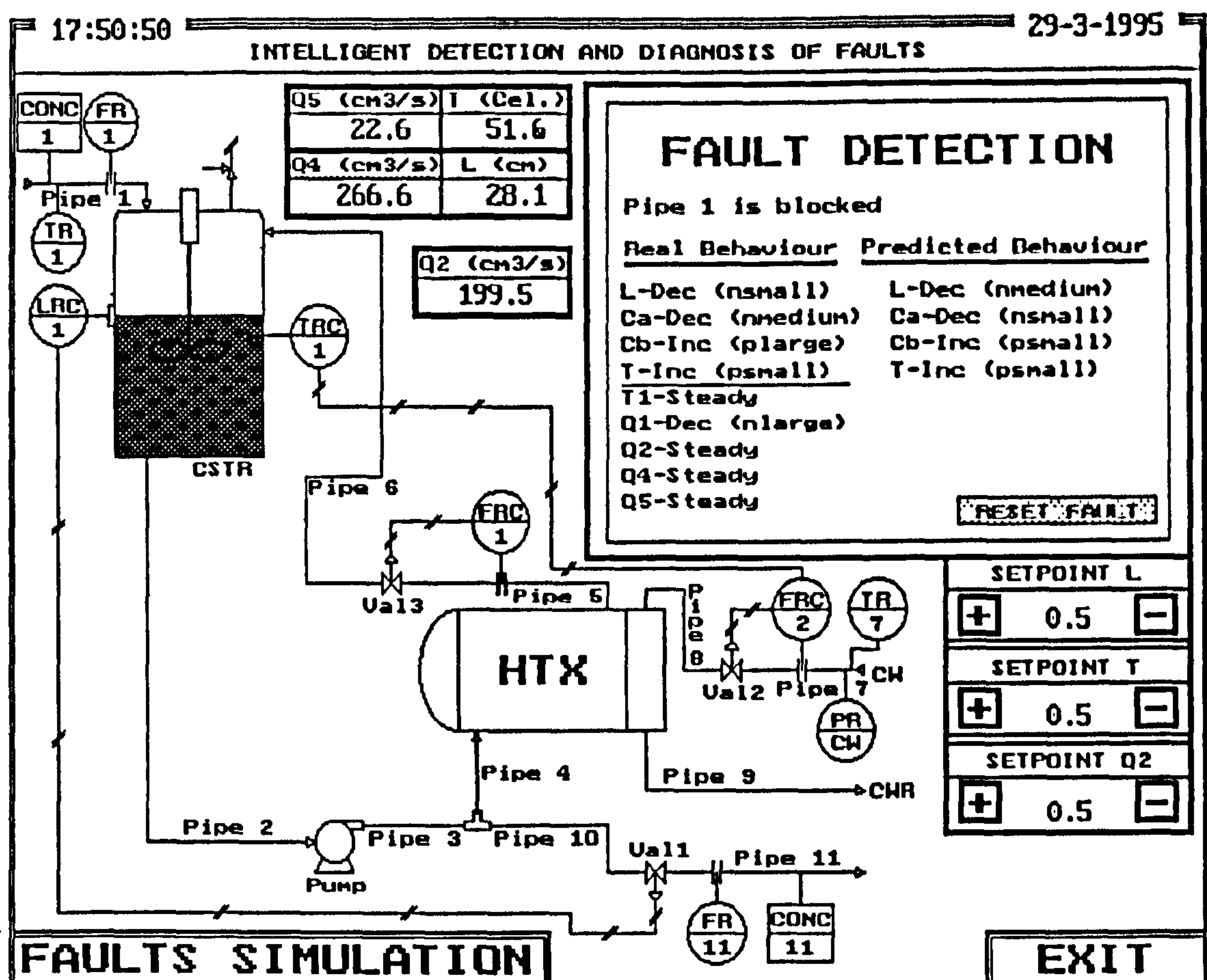


Figure 7.7 - Diagnostic of "Pipe 1 is blocked" fault

operation in the single faults space, which is represented in Table 7.2. However, for the continuous stirred tank reactor only a sub set of the AND set is considered. Table 7.4 presents all the double simultaneous abrupt faults considered, as well as their corresponding representation. The faults representation is given through the process variables used to simulate the process under a failure situation. The changes in these variables are represented in a qualitative form by the linguistic values "negative" and "positive", as described above for the single faults space.

The fault detection and diagnosis system, based on the above approach, has been applied successfully to the CSTR process where all double simultaneous abrupt faults listed in Table 7.4, have been considered. Experiments have been conducted to acquire the corresponding fault symptoms, with all the setpoints of the controlled variables taking the value 0.5. After that all the double simultaneous faults quoted above have been detected and diagnosed successfully. For instance, Figure 7.8 shows the diagnosis result after the double simultaneous faults, "Pipe 1 is blocked AND Control valve 3 fails high", have been initiated. However, let us consider the real process variables behaviour after fault, "Pipe 1 is blocked", has been initiated, and after faults, "Pipe 1 is blocked AND Control valve 3 fails high", have been initiated, respectively, as shown in Figures 7.7 and 7.8. Comparing both sets of process variables real behaviour, it can be seen that if



Faults	Representation
External feed reactant flow high AND Pipe 10 or 11 blocked or control valve 1 fails low	$[\Delta Q_1] = \text{positive}$ AND $[\Delta A_4] = \text{negative}$
External feed reactant flow high AND Control valve 2 fails high	$[\Delta Q_1] = \text{positive}$ AND $[\Delta A_5] = \text{positive}$
External feed reactant flow high AND Pipe 7, 8 or 9 is blocked or control valve 2 fails low	$[\Delta Q_1] = \text{positive}$ AND $[\Delta A_5] = \text{negative}$
External feed reactant flow high AND Pipe 4, 5 or 6 is blocked or control valve 3 fails low	$[\Delta Q_1] = \text{positive}$ AND $[\Delta A_2] = \text{negative}$
External feed reactant flow high AND Control valve 3 fails high	$[\Delta Q_1] = \text{positive}$ AND $[\Delta A_2] = \text{positive}$
External feed reactant flow high AND External feed reactant concentration too low	$[\Delta Q_1] = \text{positive}$ AND $[\Delta C_{a0}] = \text{negative}$
Pipe 1 is blocked AND Control valve 2 fails high	$[\Delta Q_1] = \text{negative}$ AND $[\Delta A_5] = \text{positive}$
Pipe 1 is blocked AND Pipe 7, 8 or 9 is blocked or control valve 2 fails low	$[\Delta Q_1] = \text{negative}$ AND $[\Delta A_5] = \text{negative}$
Pipe 1 is blocked AND Control valve 1 fails high	$[\Delta Q_1] = \text{negative}$ AND $[\Delta A_4] = \text{positive}$
Pipe 1 is blocked AND Pipe 4, 5 or 6 is blocked or control valve 3 fails low	$[\Delta Q_1] = \text{negative}$ AND $[\Delta A_2] = \text{negative}$
Pipe 1 is blocked AND Control valve 3 fails high	$[\Delta Q_1] = \text{negative}$ AND $[\Delta A_2] = \text{positive}$
Pipe 2 or 3 is blocked or pump fails AND External feed reactant temperature high	$[\Delta(\Delta P)] = \text{negative}$ AND $[\Delta T_1] = \text{positive}$
Pipe 2 or 3 is blocked or pump fails AND External feed reactant temperature low	$[\Delta(\Delta P)] = \text{negative}$ AND $[\Delta T_1] = \text{negative}$
Pipe 2 or 3 is blocked or pump fails AND Control valve 2 fails high	$[\Delta(\Delta P)] = \text{negative}$ AND $[\Delta A_5] = \text{positive}$
Pipe 2 or 3 is blocked or pump fails AND External feed reactant concentration too low	$[\Delta(\Delta P)] = \text{negative}$ AND $[\Delta C_{a0}] = \text{negative}$

Table 7.4 - Double simultaneous faults representation for the CSTR process



Faults	Representation
External feed reactant temperature high OR low AND Pipe 10 or 11 is blocked or control valve 1 fails low	$[\Delta T_1] = \text{positive OR } [\Delta T_1] = \text{negative}$ AND $[\Delta A_4] = \text{negative}$
External feed reactant temperature high OR low AND Control valve 2 fails high	$[\Delta T_1] = \text{positive OR } [\Delta T_1] = \text{negative}$ AND $[\Delta A_5] = \text{positive}$
External feed reactant temperature high OR low AND Pipe 7, 8 or 9 is blocked or control valve 2 fails low	$[\Delta T_1] = \text{positive OR } [\Delta T_1] = \text{negative}$ AND $[\Delta A_5] = \text{negative}$
External feed reactant temperature high OR low AND Control valve 1 fails high	$[\Delta T_1] = \text{positive OR } [\Delta T_1] = \text{negative}$ AND $[\Delta A_4] = \text{positive}$
External feed reactant temperature high OR low AND Pipe 4, 5 or 6 is blocked or control valve 3 fails low	$[\Delta T_1] = \text{positive OR } [\Delta T_1] = \text{negative}$ AND $[\Delta A_2] = \text{negative}$
External feed reactant temperature high OR low AND Control valve 3 fails high	$[\Delta T_1] = \text{positive OR } [\Delta T_1] = \text{negative}$ AND $[\Delta A_2] = \text{positive}$
External feed reactant temperature high OR low AND External feed reactant concentration too low	$[\Delta T_1] = \text{positive OR } [\Delta T_1] = \text{negative}$ AND $[\Delta C_{a0}] = \text{negative}$
Pipe 10 or 11 is blocked or control valve 1 fails low AND Control valve 2 fails high	$[\Delta A_4] = \text{negative}$ AND $[\Delta A_5] = \text{positive}$
Pipe 10 or 11 is blocked or control valve 1 fails low AND Pipe 4, 5 or 6 is blocked or control valve 3 fails low	$[\Delta A_4] = \text{negative}$ AND $[\Delta A_2] = \text{negative}$
Pipe 10 or 11 is blocked or control valve 1 fails low AND Control valve 3 fails high	$[\Delta A_4] = \text{negative}$ AND $[\Delta A_2] = \text{positive}$

Table 7.4 (Cont.) - Double simultaneous faults representation for the CSTR process



Faults	Representation
Pipe 10 or 11 is blocked or control valve 1 fails low AND External feed reactant concentration too low	$[\Delta A_4] = \text{negative}$ AND $[\Delta C_{a0}] = \text{negative}$
Control valve 2 fails high AND Control valve 1 fails high	$[\Delta A_5] = \text{positive}$ AND $[\Delta A_4] = \text{positive}$
Control valve 2 fails high AND Pipe 4, 5 or 6 is blocked or control valve 3 fails low	$[\Delta A_5] = \text{positive}$ AND $[\Delta A_2] = \text{negative}$
Control valve 2 fails high AND External feed reactant concentration too low	$[\Delta A_5] = \text{positive}$ AND $[\Delta C_{a0}] = \text{negative}$
Pipe 7, 8 or 9 is blocked or control valve 2 fails low AND Control valve 1 fails high	$[\Delta A_5] = \text{negative}$ AND $[\Delta A_4] = \text{positive}$
Pipe 7, 8 or 9 is blocked or control valve 2 fails low AND Control valve 3 fails high	$[\Delta A_5] = \text{negative}$ AND $[\Delta A_2] = \text{positive}$
Pipe 7, 8 or 9 is blocked or control valve 2 fails low AND External feed reactant temperature too low	$[\Delta A_5] = \text{negative}$ AND $[\Delta C_{a0}] = \text{negative}$
Control valve 1 fails high AND Pipe 4, 5 or 6 is blocked or control valve 3 fails low	$[\Delta A_4] = \text{positive}$ AND $[\Delta A_2] = \text{negative}$
Control valve 1 fails high AND Control valve 3 fails high	$[\Delta A_4] = \text{positive}$ AND $[\Delta A_2] = \text{positive}$
Control valve 1 fails high AND External feed reactant concentration too low	$[\Delta A_4] = \text{positive}$ AND $[\Delta C_{a0}] = \text{positive}$
Pipe 4, 5 or 6 is blocked or control valve 3 fails low AND External feed reactant concentration too low	$[\Delta A_2] = \text{negative}$ AND $[\Delta C_{a0}] = \text{positive}$
Control valve 3 fails high AND External feed reactant concentration too low	$[\Delta A_2] = \text{positive}$ AND $[\Delta C_{a0}] = \text{positive}$

Table 7.4 (Cont.) - Double simultaneous faults representation for the CSTR process



the diagnosis procedure is only based on the behaviour of process variables  $L$ ,  $C_a$ ,  $C_b$  and  $T$ , in the time instant  $t_i$ , it will be impossible to diagnose the double simultaneous abrupt faults. This means, since the fault, "Pipe 1 is blocked", has a quick effect on the process behaviour, using the same procedure under a double simultaneous faults situation, the diagnostic result will be "Pipe 1 is blocked".

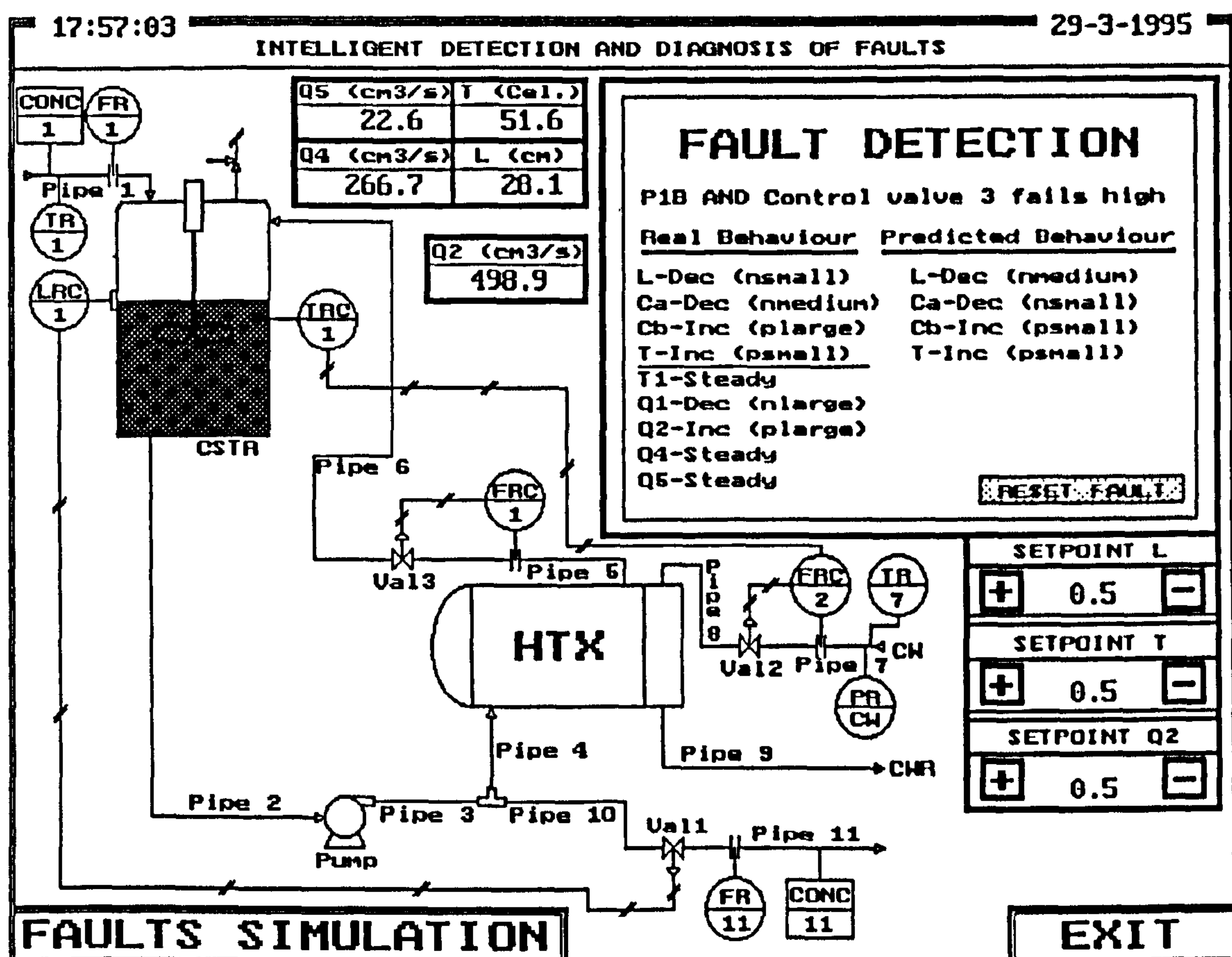


Figure 7.8 - Diagnostic of "Pipe 1 is blocked AND Control valve 3 fails high" faults

Therefore, the use of real behaviour process variables  $T_i$ ,  $Q_i$ ,  $Q_2$ ,  $Q_4$  and  $Q_5$ , in the time instant when the fault detection and diagnosis system is fired, are absolutely necessary in order to cope with multi-simultaneous faults. According to the general description of the fault detection and diagnosis system presented in sub chapter 7.3, the process variables, quoted in the last sentence, were chosen because they are directly affected by the following faults respectively:

- External feed reactant temperature high or low;
- Pipe 1 is blocked or external feed reactant flow high;
- Pipe 4, 5 or 6 is blocked or control valve 3 fails low or high;
- Pipe 10 or 11 is blocked or control valve 1 fails low or high;
- Pipe 7, 8 or 9 is blocked or control valve 2 fails low or high.

Moreover, if the diagnosis task is based on the process variables rate of change of  $T_1$ ,  $Q_1$ ,  $Q_2$ ,  $Q_4$  and  $Q_5$ , taken in the time instant  $t_1$  after a fault has been initiated, it will be impossible to diagnose the double simultaneous faults considered in Figure 7.8, because the qualitative values of  $Q_1$  and  $Q_2$  will be "*steady*" and the fault symptoms corresponding to the fault "Pipe 1 is blocked and Control valve 3 fails high" will match the fault symptoms corresponding to the fault "Pipe 1 is blocked". Hence, once again, if this procedure is followed, only the single fault with a quick effect in the process behaviour will be diagnosed.

Following the procedure described to cope with single and multiple simultaneous abrupt faults, the fault detection and diagnosis system has been successfully applied to the continuous stirred tank reactor. All the single and double simultaneous abrupt faults stated above have been initiated under different steady state conditions of the process and successfully detected and diagnosed.

## 7.6 Conclusions

This chapter has been concerned with the application of a fuzzy qualitative simulation algorithm to on-line multiple simultaneous fault detection and diagnosis. To perform the fuzzy qualitative task an approximation of the extension principle has been used, which is described in chapter 3.. This procedure has shown to be a good approximation of the results achieved through the extension principle and has the advantage that it is easier to implement.

It has been observed that the fuzzy representation of qualitative values is more general than ordinary interval representation, since it can represent not only the information stated by a well defined real interval but also the knowledge embedded in the soft boundaries of the interval. Thus, fuzzy quantity space removes, albeit not completely, the classical boundary interpretation problem, through the description of a gradual rather than an abrupt change in the degree of membership of which a physical quantity is mapped into a particular qualitative value. Moreover, to represent the semantics of the qualitative values, fuzzy numbers represented in a parametric form by 4-tuples, have been considered.

The real time expert system developed has a hierarchical structure consisting of a process simulator with control of some process variables in a lower layer and a supervisory module in an upper layer whose main aim is to detect and diagnose faults introduced in the lower layer. Therefore, in the prototype implemented, the on-line fault detection and diagnosis procedure is regarded as a supervisory task.



The current fault detection and diagnosis approach is an extension of a previous one developed to detect and diagnose single abrupt faults, which is presented in the last chapter. The system architecture of the present approach, which can cope with multiple simultaneous faults situations, has been presented in sub chapter 7.2, and the main changes performed in the previous system are discussed at sub chapter 7.3.

The present approach has been successfully applied to the mixing process, as well as to the continuous stirred tank reactor, which have been described at previous chapters. Fault detection and diagnosis of single and double simultaneous faults, which occur suddenly, have been considered. The results, achieved during experiments conducted with such processes, are presented respectively in sub chapters 7.4 and 7.5. However, it has been found that if incipient faults, which evolve gradually, are considered, the efficiency and reliability of the system is significantly affected. A further chapter will investigate the use of a fuzzy neural network coupled with a knowledge based system in an attempt to overcome this problem.

## Chapter 8

# Fuzzy Qualitative Simulation Based Fault Diagnosis with Self-Learning Abilities

### 8.1 Introduction

The extraordinary growth of artificial intelligence applications, in the last years, has been paralleled by a surge of interest in machine learning, a field concerned with the developing computational theories of learning processes and building learning machines. Because the ability to learn is clearly fundamental to any intelligent behaviour, the concerns and goals of machine learning are central to the progress of artificial intelligence.

Precise definitions of learning are hard to find, but most authorities would agree that it is a characteristic of adaptive systems which are capable of improving their performance on a problem as a function of previous experience; for example, in solving similar problems on the domain under consideration (Simon 1983).

As machine learning research has shown, learning ability manifests itself not as an all or nothing quality but as a spectrum of information processing activities, ranging from the direct memorisation of facts and acquisition of simple skills by imitation to very intricate inferential processes leading to creation of new concepts and discovery of new knowledge. It always involves a change in a system, whether human or machine, that makes it better in some sense. Efforts to develop programs exhibiting some form of learning capabilities have multiplied in recent years (Jang 1992, Saraiva and Stephanopoulos 1992, Campos and Moral 1993). A summary of some of these efforts can also be found in Michalski *et al.* (1983, 1986), Mitchell *et al.* (1986) and Forsyth *et al.* (1986).



On the basis of the results achieved so far, clearly some rudimentary machine learning abilities are possible. Already there exist programs able to formulate new concepts and discover previously unknown regularities in data; develop decision rules that can outperform human rules; draw interesting analogies; automatically learn problem-solving heuristics; or develop generalised plans for achieving a goal (Mitchell *et al.* 1986).

Introducing all required knowledge into any new knowledge based system is a very complex procedure, time consuming and error prone process, requiring special expertise. This task can be simplified by using machine learning techniques. Such techniques would enable a system to develop decision rules from examples of experts' decisions and through the automated analysis of facts in a database.

In every learning situation, the learner transforms information provided by the environment into some new form in which it is stored for future use. The nature of this transformation determines the type of learning strategy used. The basic strategies have been distinguished into learning from examples and learning by observation and discovery. The approach presented in this chapter uses the learning by observation and discovery strategy.

As pointed out by Michalski *et al.* (1983), research in machine learning encompasses three interconnected orientations:

- Theoretical analysis and development of general learning algorithms;
- The development of computational models of human learning processes, also called cognitive modelling.
- Task-oriented studies concerned with building learning systems for specific applications, also called an engineering orientation.

Clearly, the approach described below is within the third orientation. The motivation of the present study was to build a self-learning system to the fault detection and diagnosis approaches presented in the last two chapters in order to achieve improvement in the reliability of the systems. So, we are in the engineering orientation domain.

Since diagnosis is a dominant application area of expert systems, the ability of learning would be a desirable property for a fault diagnosis system. Several fault diagnosis systems with learning properties have been reported (Pazzani 1986, 1987, Rich and Venkatasubramanian 1989, Zhang and Roberts 1991b). They are called failure-driven learning diagnosis systems because, as in the approach described here, learning is initiated when a failure occurs in the diagnosis system. Other examples of failure-driven self learning systems have been reported by Sussman (1975) and Charniak *et al.* (1985).



In the first three approaches quoted above, fault diagnosis is based on a set of heuristic rules, which are believed to give reliable diagnosis. These heuristic rules are in the form:

- *IF* Antecedent *THEN* Consequence.

The heuristic rule antecedent consists of fault symptoms, which are linked by logical operators such as *AND* and *OR*, while the rule consequence represents the corresponding fault. Therefore, when the behaviour of the measurement process variables match the antecedent of a rule, the rule is fired and the corresponding fault is diagnosed. However, since the heuristic rules may not be perfect, a failure may occur during the diagnosis task, such that the hypothetical fault proposed by a rule is incorrect. Once such a failure has occurred, the heuristic rule generating the wrong hypothesis is modified and a new rule is generated. The task of learning is carried out based on a deep model of the system being diagnosed. The failed heuristic rule is modified by including additional features in its antecedent part, which are obtained from reasoning through the deep model, such that its applicability is limited and will not be fired in future similar situations. On the other hand, when a fault occurs at the first time in the process under consideration a new heuristic rule corresponding to a newly discovered fault from reasoning through the deep model is added.

However, since when considering complex processes the diagnosis result is usually obtained by the chaining of a set of rules, when a failure occurs in the fault diagnosis system, it may not be easy to decide which specific rule is responsible for the failure. Therefore, the method described above may not be applied in a straight forward manner for fault diagnosis of complex plants.

In the self-learning fault diagnosis system proposed by Zhang and Roberts (1991b), a methodology to adjust threshold values, which is responsible for firing a fault detection and diagnosis system based on a deep qualitative model of the process being monitored, is pointed out. In this approach, the self-learning system through reasoning its own behaviour will find any inappropriate parameters and suggests correct ones. It has been observed that in fault detection and diagnosis systems where threshold values are used for firing such a system, incorrect settings of such parameters could lead to a wrong fault detection and diagnosis (Iri *et al.* 1979).

In the self-learning on-line fault detection and diagnosis described in this chapter, the learning task is carried out differently from above. However, this approach shares some properties with the systems quoted. The current self-learning system has been coupled with the on-line fault detection and diagnosis systems described in the previous two chapters, which detected and diagnosed faults based on fuzzy qualitative reasoning. The methodology followed in the present



system, which use inductive and deductive learning techniques, is described in the next sub chapters.

Sub chapter 8.2 presents the system architecture. In sub chapter 8.3 a methodology to tune fuzzification parameters is described. A method for acquiring fault symptoms from on-line measurement data is presented in sub chapter 8.4. Some results achieved during simulation studies conducted with the mixing process described in chapter 4. and with the continuous stirred tank reactor described in chapter 5., are presented in sub chapter 8.5. The last sub chapter contains some concluding remarks.

## 8.2 System Architecture

Since the ability of learning would be a desirable property for a fault detection and diagnosis system, the fault detection and diagnosis approaches described in the last two chapters have been extended for possessing such an ability.

Hence, according to Figures 6.1 and 7.1, which depicted the architecture of the fault detection and diagnosis systems, respectively, with single and multiple faults detection and diagnosis capabilities, a new module has been developed for the upper layer of the overall computational systems. The aim of this module is to provide the fault detection and diagnosis approaches quoted above with self-learning abilities. Moreover, since the kinds of failures in the approach are analogous, the same self-learning module has been applied for both architectures. Therefore, the self-learning fault detection and diagnosis system proposed in this chapter, has the architecture depicted in Figure 8.1.

The upper layer of the computational system developed, which consists of the self-learning fault detection and diagnosis system proposed in this chapter, can be viewed as a hierarchical fault diagnosis system with two levels. The lower level consists of an on-line supervisory fault detection and diagnosis system, which can be any one of those described in the last two chapters, and the upper level consists of a self-learning system which has the ability for reasoning the behaviour of the lower level if it failed to give a correct result.

Therefore, when the on-line fault detection and diagnosis system fails to give the location of a hypothetical detected fault in the process under concern, the self-learning module is triggered. There are two kinds of such failures: one is that the system has perceived that a fault or faults occur in the process but really no fault occurred; another one is that a new fault occurred in the process. The second failure quoted includes the situation where a fault has been occurring in the process, but not with the present steady state conditions. Note that the fault symptoms of a specific fault or faults depend on the steady state conditions of the process.

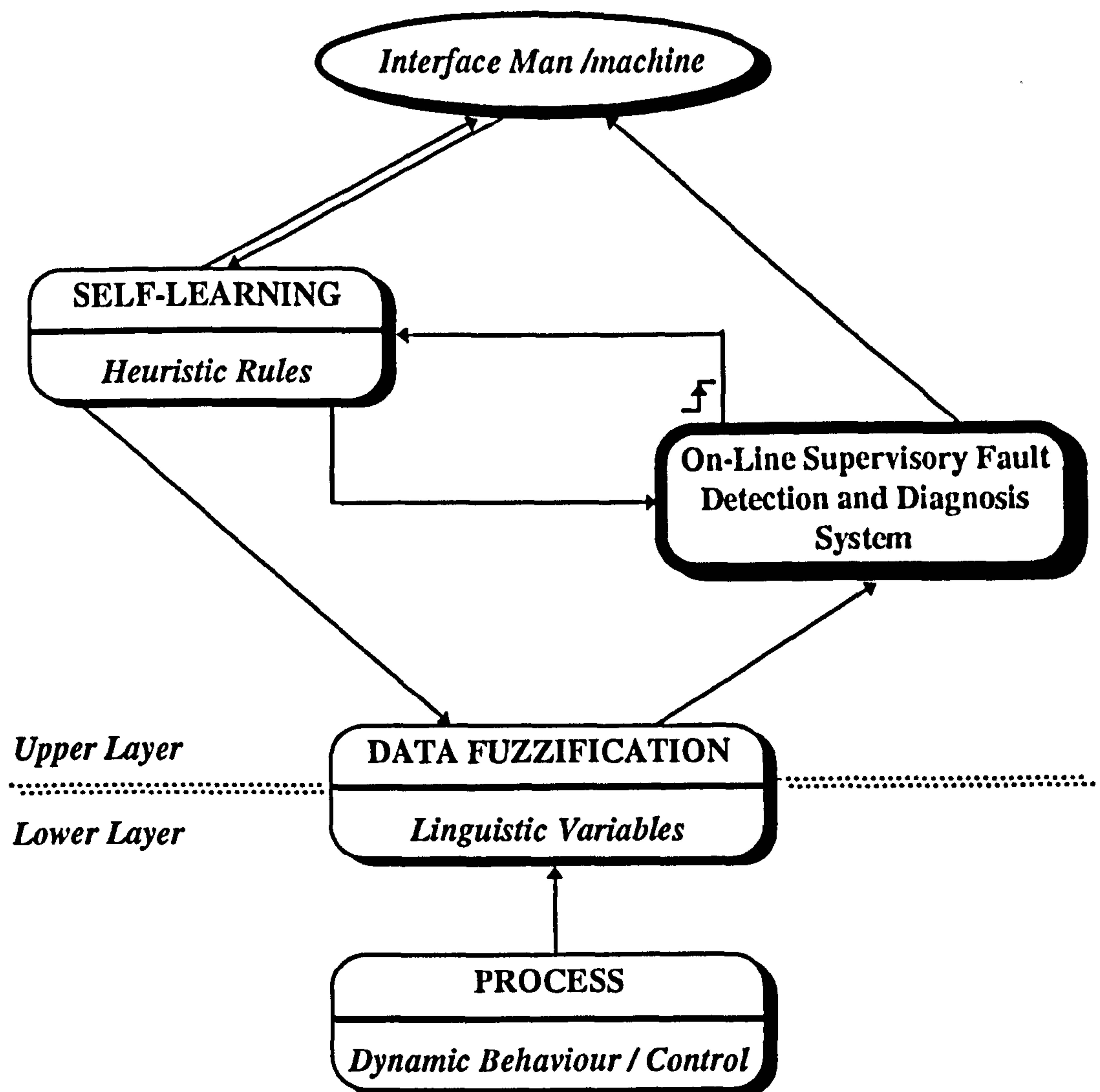


Figure 8.1 - General scheme of a computer assisted fault diagnosis system  
with self-learning abilities

However, before the self-learning system is triggered the computational system developed requests the operator to identify which kind of failure has occurred in the fault detection and diagnosis tasks. This procedure is performed through an iterative menu as depicted in Figure 8.2 for simulation studies conducted with the mixing process, which is described in chapter 4.. An analogous menu has also been used in simulation studies conducted with the continuous stirred tank reactor, which is presented in chapter 5..

The learning system developed is based on a hybrid inductive and deductive learning approach. Inductive learning is used to hypothesise causal relations and is supported by general knowledge of the forms of causal relations in terms of functional dependencies between quantities. Deductive learning is used to hypothesise and verify causal relations and is supported by knowledge of the kinds of causal mechanisms which exist in the domain of physical systems.

Moreover, as quoted above in every learning situation, the learner transforms information provided by the environment into some new form. If the transformation process involves



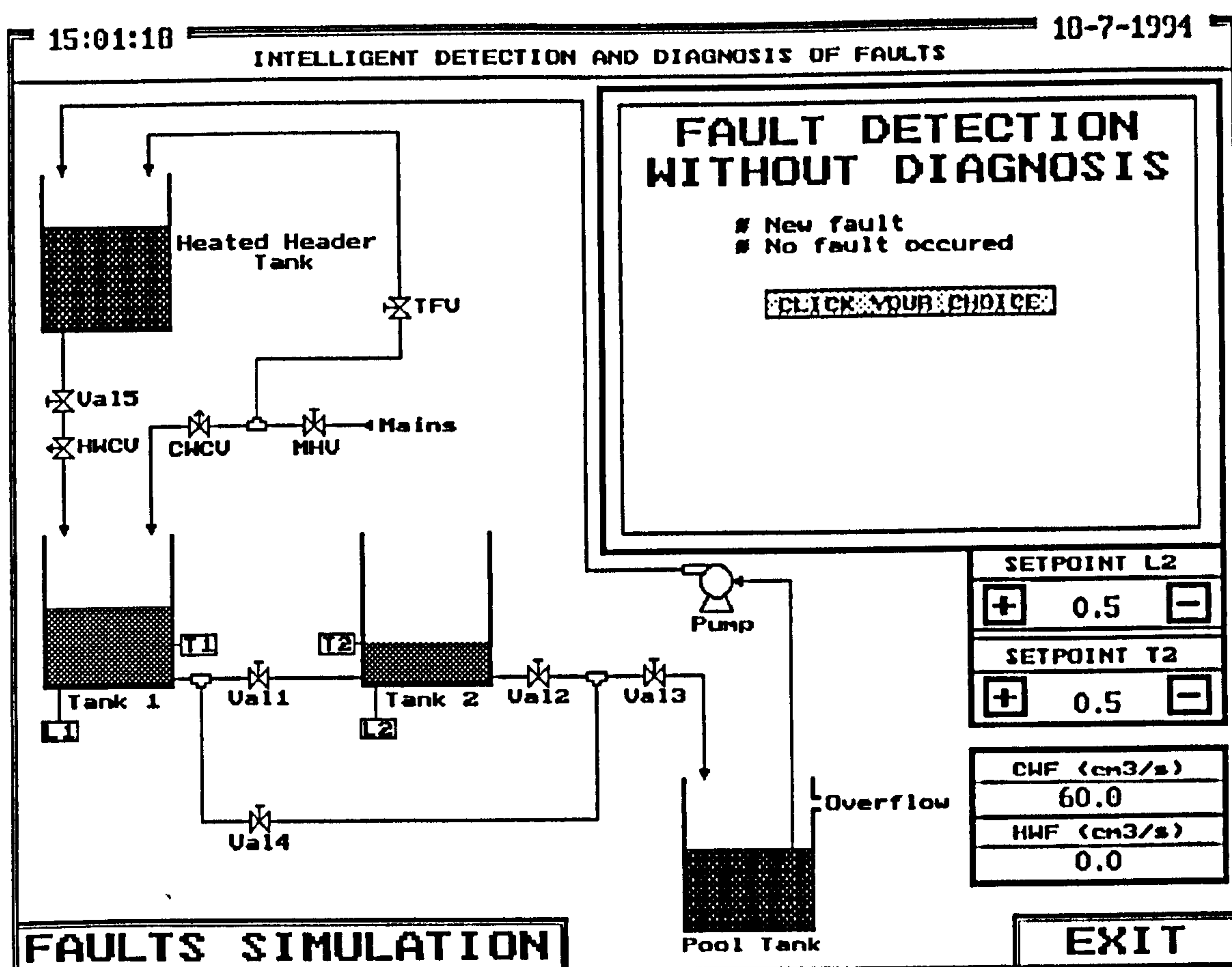


Figure 8.2 - Interface man/machine when the diagnosis system fails

generalisation of input information and selection of the most plausible or desirable result, that is called in the machine learning literature by inductive inference, and then we have inductive learning. Deductive learning includes knowledge reformulating, knowledge compilation, creation of macro-operators, caching, chunking, etc. (Michalski 1986). In the self-learning fault detection and diagnosis system presented here, deductive learning includes knowledge compilation.

The next two sub chapters describe the reasoning behind the inductive and deductive learning abilities associated with the current self-learning fault detection and diagnosis system.

### 8.3 Adjusting Fuzzy Membership Functions Through Inductive Learning

The inductive learning technique has been successfully applied in a wide range of areas from medical disease prediction to financial market forecasting, from fault diagnosis of printed circuit boards to predicting problems with the application of surface mount adhesives (Donald

1994). In this paper a wide range of engineering applications, such as condition monitoring, quality control, fault tolerant systems, image inspection, etc., are also reported.

A particularly interesting application of this technique is described by Michie (1982). He considers the possibility of constructing an expert system backwards because it is easier that way. An expert system can be considered as a function from the input space to the expert solution. However, this forward transformation is typically difficult, but sometimes the reverse transformation, the inverse function, from a given solution to the problem that would have generated it, is easily computable. In the development of an expert system for soybean disease diagnosis, Michalski and Chilansky (1980) inductively derived the diagnostic rules from a collection of symptoms and the diagnosed disease.

In the present approach when an unsuccessful diagnosis is achieved due to the first kind of failure quoted above for the fault detection and diagnosis system, an inductive learning technique is used in order to avoid such a future similar situation, automatically. The main reason for this kind of failure is incorrect settings of normalisation parameters which are used to make the normalisation of the changes in the process variables, in the range  $[-1, 1]$ . The linguistic threshold values, which are represented in a form of fuzzy sets such as *psmall* or *nsmall*, that are responsible for triggering the fault detection system as quoted above, are directly affected by these normalisation parameters. Therefore, they will dramatically affect the performance and reliability of the fault detection and diagnosis system.

When a false fault is detected by the reason mentioned above and afterwards confirmed by the operator, through the interface depicted in Figure 8.2, that no fault has actually occurred in the process, the self-learning system is fired. Then the system begins to investigate its own behaviour in order to detect which parameter (or parameters) was responsible for triggering the fault detection system. After these parameters are identified, the system starts to adjust their fuzzy membership functions, in order that in a future similar situation the fault detection system is not triggered. The methodology developed is composed of the following steps:

1. Find the real value of the process variable change.
2. For this process variable evaluate a new normalisation parameter, such that the linguistic value of the process variable achieved through the data fuzzification system would be *psmall* or *nsmall*.

This procedure has been successfully applied during simulation studies conducted with the mixing process, as well as with the continuous stirred tank reactor. Some results were obtained which are presented in sub chapter 8.5.



# 8.4 Learning Fault Diagnosis Heuristic Rules

The deductive learning ability of the self-learning system is used to compile new knowledge about fault symptoms. When a fault or set of faults occurs at the first time with the present steady state conditions of the process under consideration, the fault detection and diagnosis system requests the operator to confirm that a fault has occurred in the process and afterwards to give a description of that fault or faults. An example of such an interface with the operator is depicted in Figure 8.3.

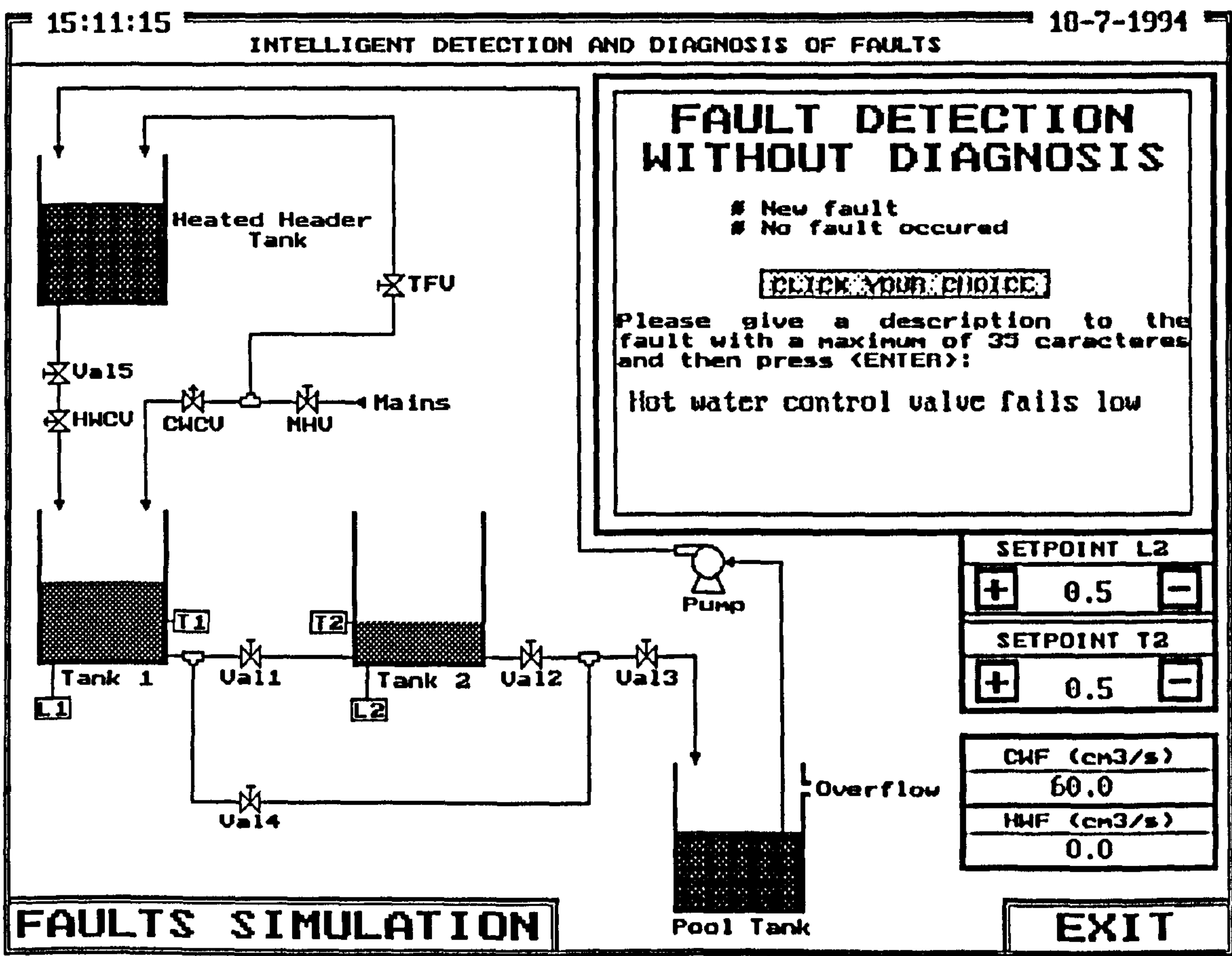


Figure 8.3 - Interface man/machine when a new fault or faults occur in the process

Therefore, the self-learning system starts to investigate the real behaviour of the process variables, according to the diagnosis reasoning procedures described in the previous chapters. This strategy provides values in a linguistic form whose semantics are also represented by fuzzy numbers and stored in the knowledge base in order to be used for possible diagnosis of that fault or set of faults the next time that it occurs. This procedure allows us to build the knowledge base on-line very efficiently, simplifying the task of introducing all the fault symptoms in the knowledge base. Moreover, since fault symptoms are acquired on-line, following the proposed procedure, there



is no need for a human expert in the domain to build the knowledge base. By this means, the fault detection and diagnosis system performance will improve over time.

To test this ability of the self-learning system several studies have also been conducted with both processes quoted above. The results shown in the next sub chapter demonstrate the successful application of this procedure for building on-line the fault diagnosis knowledge bases of the mixing process and the continuous stirred tank reactor.

## 8.5 Case Studies

Both fault detection and diagnosis systems described in the last two chapters have been used to test the self-learning fault detection and diagnosis system proposed in this chapter. Moreover, simulation studies with both processes previously presented, which are the mixing process and the continuous stirred tank reactor, have been conducted. All the single abrupt faults and all the double simultaneous abrupt faults previously mentioned for such processes, have been considered during the current simulation studies. Successful results have been obtained which are presented in the remainder of this sub chapter.

Since the fault detection and diagnosis systems implemented for both processes, and described in the last two chapters, are well tuned, all the normalisation parameters have been set appropriately. Therefore, to test the inductive learning ability of the self-learning system, which is described above, initially it is required to deviate some normalisation parameters values from their pre-set values. Figure 8.4 shows the result of one such experiment conducted with the mixing process.

After the self-learning system has been developed, the fault diagnosis knowledge bases have been built on-line very efficiently. All the single and double simultaneous abrupt faults mentioned above were separately initiated and the corresponding fault symptoms have been acquired for different steady state conditions of the process under consideration. As illustrated by the following figures successful results have been achieved. Figure 8.5 presents an example of knowledge acquisition for a single abrupt fault situation, which has been achieved during simulation studies conducted with the mixing process. This example has been obtained by using the on-line fault detection and diagnosis system proposed in chapter 6.. Figure 8.5 shows the fuzzy qualitative values corresponding to the real behaviour of the process variables, which are used to perform the diagnosis task, under the single abrupt fault, "Hand valve 1 is blocked", with the setpoints of the controlled variables having the values 0.5. It is worth noting that if the controlled variables had taken other values the fault symptoms for such a fault could be slightly different.



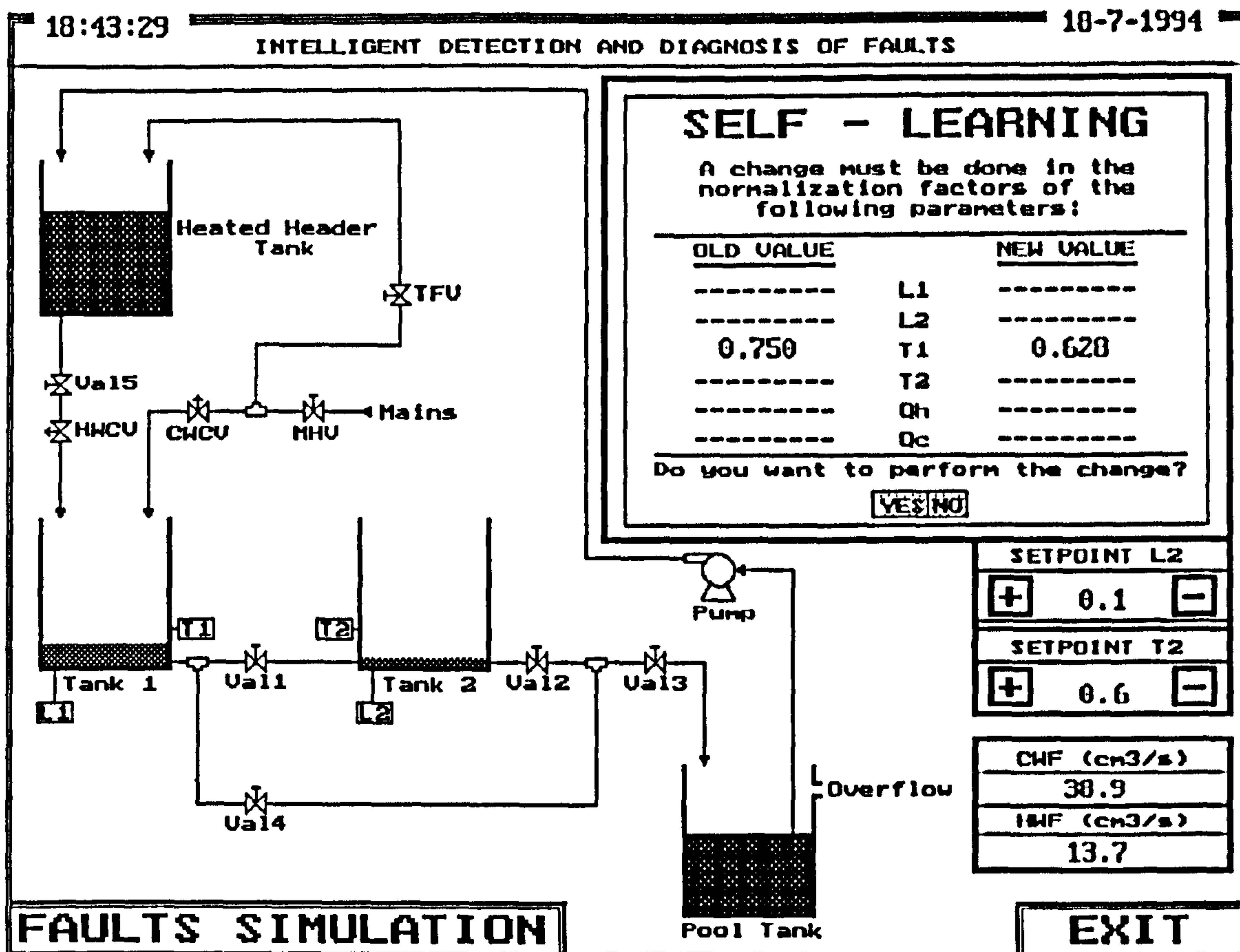


Figure 8.4 - Learning the adjustment of fuzzy membership functions

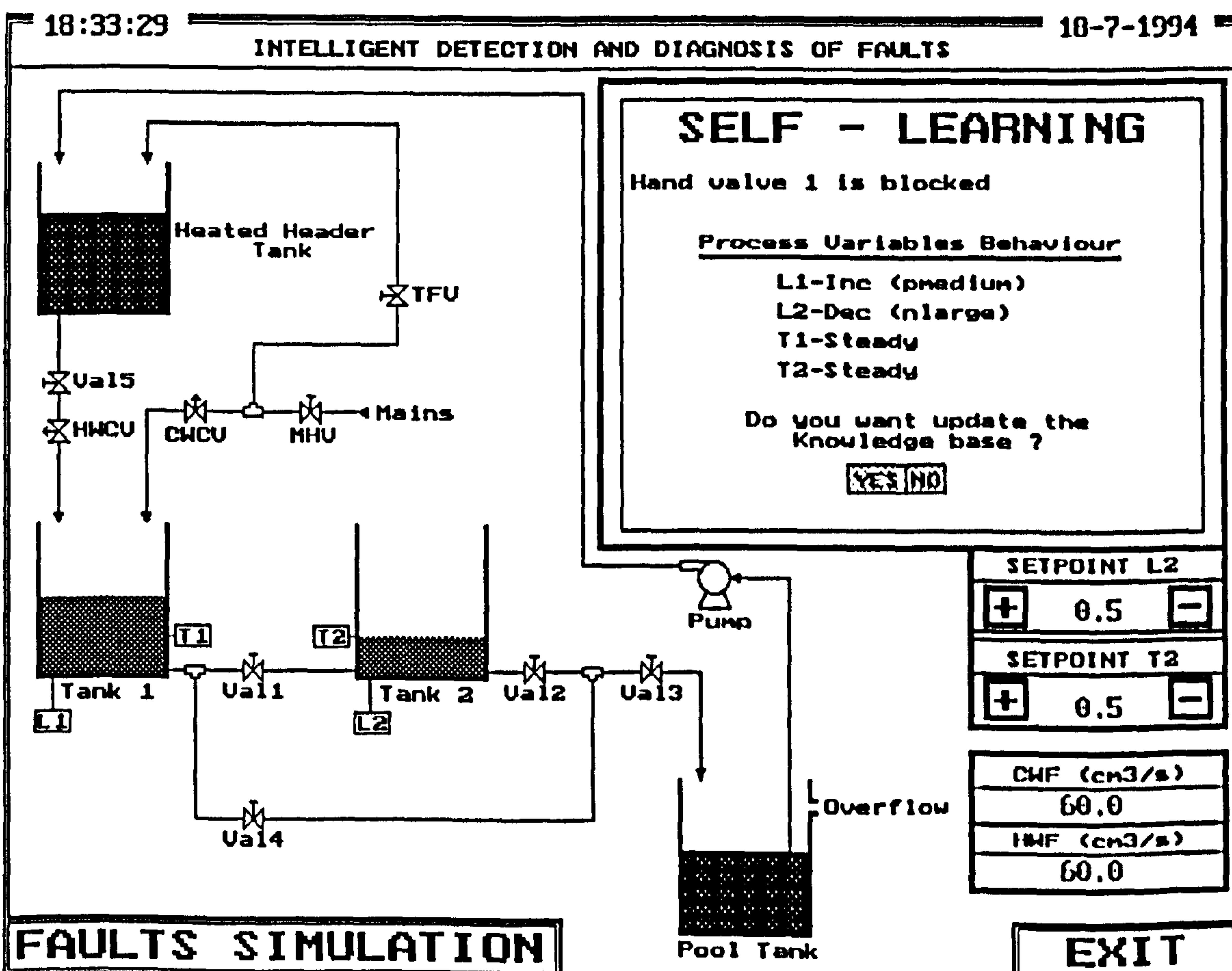


Figure 8.5 - Self-Learning single abrupt fault symptoms in the mixing process

A similar situation for a double simultaneous abrupt fault in the continuous stirred tank reactor is presented in Figure 8.6. For this situation the on-line fault detection and diagnosis system, described in chapter 7., has been used. According to the description given in this chapter and, in contrast with the example shown in Figure 8.5, it can be seen from Figure 8.6 that in this example two time instants are considered for acquiring the fault symptoms. This means that a fault or faults are detected in the time instant,  $t$ , but can only be diagnosed in the time instant,  $t+1$ .

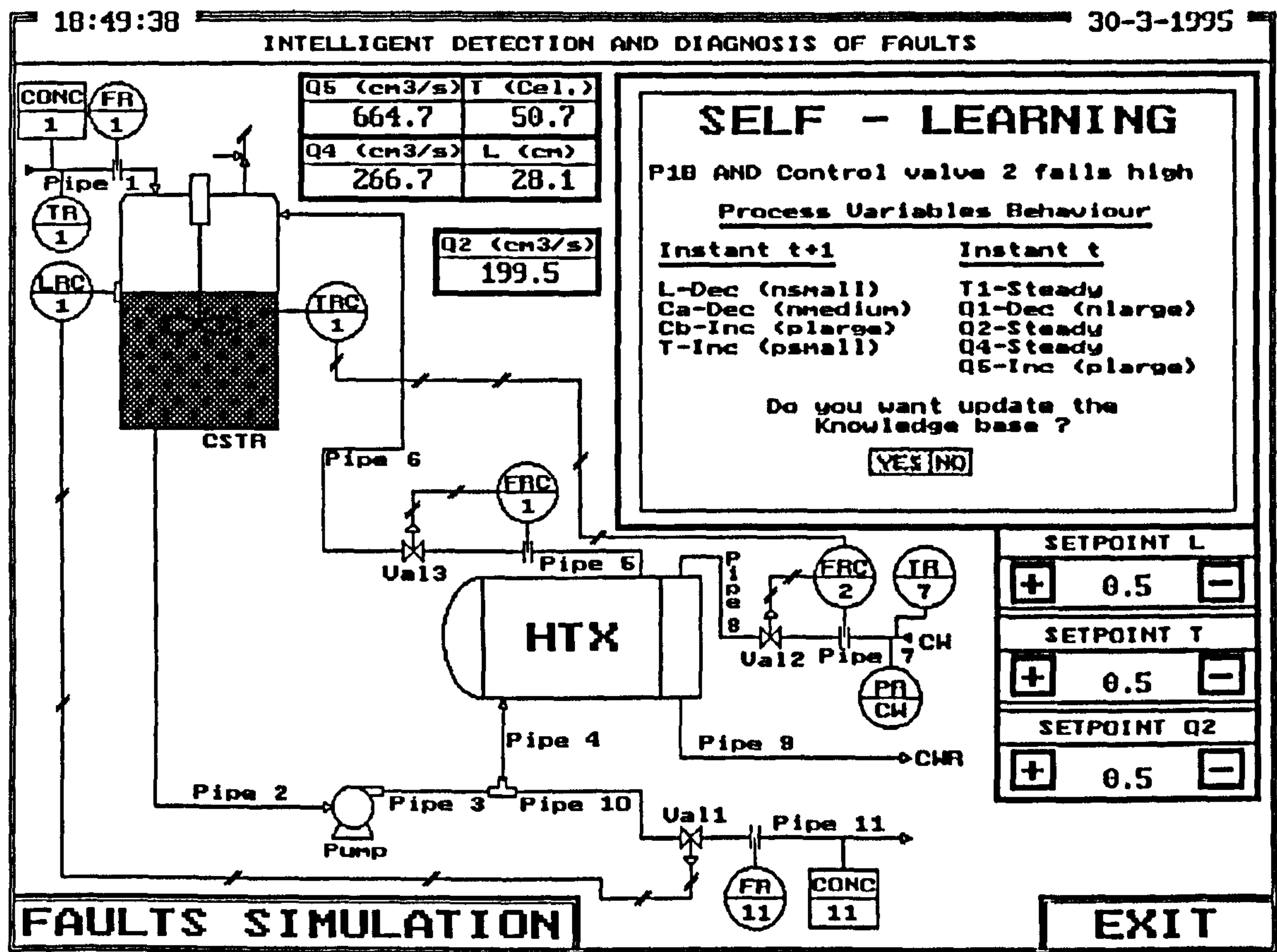


Figure 8.6 - Self-Learning double simultaneous abrupt faults symptoms in the CSTR

## 8.6 Conclusions

In this chapter an on-line self-learning fault detection and diagnosis system is investigated. The ability for reasoning about its own behaviour could make a knowledge based system more intelligent and autonomous. The self-learning system developed can be understood as a diagnosis system working in an upper level, which will reason about the fault detection and diagnosis system working in a lower level, when any undesirable performance occurs there. By such means, any inappropriate parameters in the fault diagnosis system could be found and updated. Therefore, the computational system developed possesses adaptive properties.



A method for learning fault symptoms from on-line sampled data has also been investigated. By using machine learning techniques, fault symptoms in the form of heuristic rules can be automatically acquired, and this eases the knowledge acquisition task. Through self-learning of fault symptoms, the diagnosis system can gradually improve its performance in terms of diagnostic efficiency. This enhances diagnostic reliability and can cover a wide range of potential faults.

The overall computational system implemented has been tested during simulation studies conducted with a mixing process, as well as with a continuous stirred tank reactor. The detection and diagnosis of single and double simultaneous abrupt faults have been considered for evaluating the performance and reliability of the on-line self-learning fault detection and diagnosis approach proposed in this chapter. Successful results have been obtained.

## **Chapter 9**

# **Generating Fault Detection Heuristic Rules Through Shallow and Deep Knowledge of the Process**

### **9.1 Introduction**

During the last few years there has been a dramatic increase in the number of expert system applications. As a matter of fact, any one can find huge numbers of reported applications in the periodicals and conference proceedings of many subjects. Early expert systems, such as MYCIN (Harmon and King 1985, Jackson 1986) and DENDRAL (Johnson and Keravnou 1984), contain empirical knowledge of experts in their domains. Many recent expert systems contain knowledge which may not necessarily be experience of some experts, and are also called knowledge based systems. The terms "expert systems" and "knowledge based systems" are used interchangeably in some artificial intelligence literature (Harmon and King 1985). However, in the remainder of this report the author will use the term knowledge based system.

A number of knowledge based systems have been reported, which perform fault detection and diagnosis by the method of heuristic classification. In this method, diagnostic knowledge is represented mainly in terms of heuristic rules, which perform a mapping between data abstraction (typical symptoms) and solution abstraction (typical disorders). Such a representation is sometimes called "shallow" because it does not contain much information about the causal mechanisms underlying the relationship between symptoms and faults. The rules typically reflect empirical associations derived from experience, rather than a theory of how the device under diagnosis



actually works. The latter knowledge, used in conception of knowledge based systems, is sometimes called "deep" knowledge because it involves understanding the structure of the device and the way its components function (Jackson 1990).

Thus, knowledge based fault detection and diagnosis systems can be divided into shallow knowledge or deep knowledge approaches according to the nature of the knowledge employed to build up the knowledge base. For instance, MYCIN, the knowledge based system quoted above, is a typical shallow knowledge based medical diagnosis system capable of handling uncertain information. The knowledge is represented by heuristic rules and, quite often, fuzzy reasoning is used since the knowledge is frequently uncertain. Knowledge acquisition is the key task associated with the shallow knowledge based systems. Expertise covering a wide range of problems must be encoded into the knowledge base. The task of knowledge acquisition is very time consuming since the process operators may know little about knowledge engineering, and therefore the interchange of information between a knowledge engineer and a process operator may not be carried out efficiently. This issue is often referred to as the "knowledge bottle neck" (Moor and Kramer 1986, Price and Lee 1988). Moreover, in a industrial process, many faults needing to be diagnosed may never have been experienced and, for new or recently developed plants, there may be little applicable experimental knowledge. Due to these drawbacks, in recently developed knowledge based systems, shallow knowledge supplements knowledge based schemes. Several knowledge based approaches which combine deep knowledge with shallow knowledge have been reported (Swartout 1983, Kahn 1988, Venkatasubramanian and Rich 1988).

The so called deep knowledge includes models of the process under concern and fault models of different process units. The models can be in the form of a set of numerical equations, or a qualitative model, or even in the form of rules compiled from a model. Fault detection and diagnosis systems based on any of these models can be called deep knowledge based systems (Scarl *et al.* 1987).

The heuristic rules used in conception of shallow knowledge based approaches lack process generality and they tend to fail under novel circumstances. Recently reported knowledge based fault detection and diagnosis systems use the deep knowledge based approach or use a combined approach where deep knowledge plays a dominant role. The advantages of deep knowledge based approaches are that they can provide reliable behaviour for infrequently occurring faults, and some deep knowledge is general in nature and can be used for other processes.

However, as Clancey (1985) has pointed out, the deep models required by the latter deep knowledge based systems are hard to construct, even for relatively simple electronic devices. Even Genesereth (1984) acknowledges that "not all design descriptions are tuned for the task of diagnosis"; although part of the motivation for using design descriptions is surely that they are not



supposed to need tuning unlike heuristic rule sets. Therefore, in order to narrow the diagnosis focus in the process under consideration, as well as in order to facilitate the process variables behaviour analysis, some methodologies have been pointed out by several researchers. A fault detection and diagnosis system based on a deep knowledge approach, which tries to explore the causal path from the observed abnormalities to their causes and, hence, locate any associated fault, is reported by Moor and Kramer (1986). Another methodology is proposed by Finch and Kramer (1988). In their approach, an industrial process is decomposed into several subsystems according to their functions and then diagnosis is performed by identifying the source system which is the subsystem where the fault occurs. Further the fault is located in the source system. A similar approach is pointed out by Steels (1989), but in this approach the function of the system being diagnosed is hierarchically decomposed. Zhang and Roberts (1991a) have proposed a fault diagnosis system based on structural decomposition of the process under concern and component functions.

However, most reported fault detection and diagnosis systems only deal with a single failure assumption. Moreover, in most of the reported approaches, in order to avoid false diagnosis under transient process behaviour, the diagnosis system trigger is based on threshold values or simply switch off when a setpoint change is performed. These threshold values will affect the performance of the diagnosis system and should be set properly according to previous operational experience of the process under consideration. This issue has been discussed in previous chapters, as well as by Zhang (1991).

In this research, a systematic methodology for generating fault detection heuristic rules is proposed. The goal is to develop a knowledge based fault detection system with the abilities to cope with multiple fault situations and increase the system reliability under transient behaviour situations. Fault detection heuristic rules are generated from knowledge of system structures and component functions. Deep and shallow knowledge will be combined and such a system will be used to trigger a fault diagnosis system based on a fuzzy neural network, which is presented in the next chapter. There, the performance and reliability of the overall fault detection and diagnosis approach are analysed, where single and double simultaneous faults are considered as abrupt and incipient faults.

This chapter is organised in the following sub chapters. Sub chapter 9.2 provides a methodology for description of system structures. In sub chapter 9.3 the reasoning for generating fault detection heuristic rules is described. The application of such a methodology is applied to the mixing process presented in chapter 4., as well as to the continuous stirred tank reactor described in chapter 5., respectively, in sub chapters 9.4 and 9.5. In sub chapter 9.6 some concluding remarks are presented.



## 9.2 Description of System Structures

In order to facilitate the behaviour analysis of process variables, the process under concern is structurally decomposed into several subsystems, where the structural decomposition corresponds to the plant topology.

By using a graph similar to the Signed Directed Graph (SDG) (Iri *et al.* 1979, Oycleye and Kramer 1988), the process can be represented by a graph which contains nodes and directed arcs. Each node represents a subsystem and the directed arcs represent interactions between subsystems. For instance, if a hypothetical system is divided into three subsystems,  $S_1$ ,  $S_2$ , and  $S_3$ , where each one interacts with each other, we can represent such a system by a Directed Graph shown in Figure 9.1.

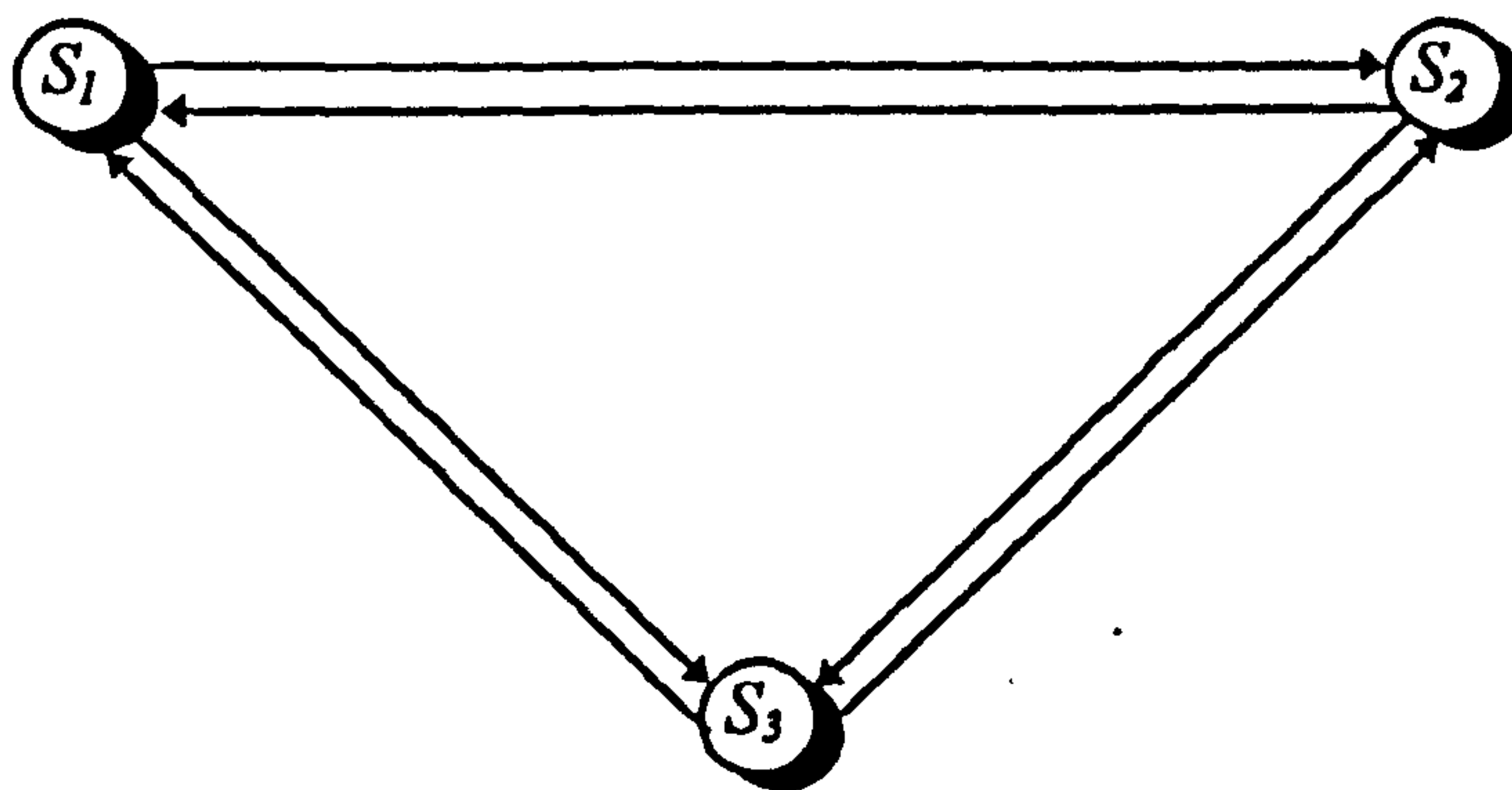


Figure 9.1 - A directed graph

Moreover, a Connection Matrix,  $C$ , can be used to represent the interactions between the subsystems. If the process is decomposed into  $n$  subsystems, then the Connection Matrix for such a system is a  $n \times n$  matrix. The element of  $C$ ,  $c_{ij}$ , is defined as follows,

$$c_{ij} = \begin{cases} 1, & \text{if subsystem } S_i \text{ can directly affect subsystem } S_j. \\ 0, & \text{otherwise.} \end{cases} \quad (9.1)$$

Since a subsystem can affect itself, according to equation (9.1) the diagonal elements of the Connection Matrix are all ones.

The state of a system is described by its measurements and a subsystem is abnormal if one of its measurements is abnormal, where such a situation can be represented by the following equation,

$$AB(S_i) \Leftarrow \exists k, k \in \overline{1, m_i}, AB(m_{ik}) \quad (9.2)$$

which states, that if there exists in subsystem  $S_i$  a measurement,  $m_{ik}$ , which is abnormal, then subsystem  $S_i$  is abnormal. In equation (9.2) the following notation is used,

$AB$  - is a predicate meaning abnormal;

$m_i$  - is the total number of measurements in  $S_i$ ;

$m_{ik}$  - is the  $k$ th measurement in  $S_i$ .

In the Connection Matrix, if the element  $c_{ij}$  takes the value one, then subsystem  $S_i$  can affect subsystem  $S_j$ . Hence, this means that at least one of the process variables in  $S_i$  can affect those in  $S_j$ . The Connection Matrix only provides a rough description on the relationships among subsystems. A refined description can be given by the Measurement Causal Matrix,  $CM_{ij}$ . If there are  $n$  measurements in  $S_i$  and  $m$  measurements in  $S_j$ , then the Measurement Causal Matrix between  $S_i$  and  $S_j$ ,  $CM_{ij}$ , is a  $n \times m$  matrix. The element of  $CM_{ij}$ ,  $cm_{ij}^{kl}$ , is determined through the following equation,

$$cm_{ij}^{kl} = \begin{cases} 1, & \text{if the } k\text{th measured variable in } S_i \text{ can directly} \\ & \text{affect the } l\text{th measured in } S_j. \\ 0, & \text{otherwise.} \end{cases} \quad (9.3)$$

Causal relationships also exist between measured variables within a subsystem. The Self-Causal Matrix,  $CS_i$ , is responsible for representing these relationships. If there are  $n$  measurements in subsystem  $S_i$ , then the Self-Causal Matrix for subsystem  $S_i$  is a  $n \times n$  matrix. Each element of  $CS_i$ ,  $cs_i^{kl}$ , is determined according to the following equation,

$$cs_i^{kl} = \begin{cases} 1, & \text{if the } k\text{th measured variable in } S_i \text{ can directly} \\ & \text{affect the } l\text{th measured variable in } S_i. \\ 0, & \text{otherwise.} \end{cases} \quad (9.4)$$

Since a measurement can affect itself, according to the last equation the diagonal elements of the Self-Causal Matrix are all ones.



The Directed Graph together with the above defined matrices give a description of the process under consideration. Fault detection heuristic rules can be generated from this description. This procedure is described in the next sub chapter.

### 9.3 A Methodology for Generating Fault Detection Heuristic Rules

The reasoning behind the development of the fault detection heuristic rules is based on the predicate stated by equation (9.2). Therefore, let us consider that the  $j$ th measurement in the  $i$ th subsystem presents an abnormal behaviour, which according to equation (9.2) is represented by  $AB(m_{ij})$ . Then a search is conducted to causally look for any measured variable in subsystem  $S_i$  which could be responsible for the observed abnormality in  $m_{ij}$ . If such a variable exists, then it is retained and a fault detection heuristic rule must be generated. At this stage, this search is guided by the Self-Causal Matrix of subsystem  $S_i$ . Similar searches are also performed to find further causes in  $S_i$  for the retained variable. If there is another variable in subsystem,  $S_i$ , which can directly affect the retained variable behaviour, then this one is also retained and another diagnostic rule is generated.

If there are no more variables in  $S_i$  which could be responsible for the observed abnormality, then the causal search at subsystem  $S_i$  is terminated. Therefore, a search is conducted to find all the subsystems that are connected with the subsystem  $S_i$  which can directly affect any measurement variable in  $S_i$ . These subsystems form the following set,

$$\{\forall j S_j, c_{ij} = 1, j \neq i\} \quad (9.5)$$

Thus, in the procedure followed for generating the above set, the Connection Matrix plays the main role. The goal is to obtain all the subsystems whose measurement variables can directly affect the measurement variables in the subsystem where such an abnormal behaviour is observed.

Next, a search is conducted through all the subsystems that compose the above set in order to find all the measured variables in other subsystems, which could directly affect the last retained variable. If such variables exist, then other heuristic rules are generated. At this stage the search procedure is guided by the Measurement Causal Matrix, which gives us the detailed information about interactions between subsystems. Once the search procedure is terminated, certain process shallow knowledge, in the form of some specific heuristic rules, can be used. By this manner, the reliability of the fault detection system can be increased.



The knowledge base of the knowledge based fault detection system will be built up with the fault detection heuristic rules achieved following the procedure described above. To fire the rules an inference engine with forward chaining abilities is used. Therefore, the fault detection heuristic rules will be chained in a forward manner and then, when the behaviour of the measurement variables match the antecedent parts of a heuristic rule, a fault detection enable flag will be settled. This flag will be used to trigger a fault diagnosis system, in order to locate the fault or faults in the process under consideration. In contrast with the on-line fault detection and diagnosis systems described in previous chapters, the main advantage of the present approach is that the fault detection task is not based on threshold values. Hence, transient behaviours of the process under concern can be considered as well as incipient faults whose development occurs gradually, instead of suddenly as considered for abrupt fault situations in previous chapters.

A benefit of the rule based format is that the fault detection rules can be augmented by any available heuristic knowledge about a particular process. The procedure, which has just been introduced for generating fault detection heuristic rules, has been applied to a simulated mixing process and to a simulated continuous stirred tank reactor. The development of these heuristic rules is described in the next two sub chapters, respectively.

## 9.4 Formulation of Fault Detection Rules for the Mixing Process

The mixing process, described in chapter 4., has been used as a test bed of the methodology presented in the previous sub chapters. This process is decomposed into two subsystems. The first subsystem includes the following components: hot and cold water control valves, tank 1 and the associated sensors. Components of the second subsystem are hand valves 1 and 2, tank 2 and associated sensors. The Directed Graph corresponding to this decomposition is shown in Figure 9.2, from which it can be seen that the two subsystems can affect each other. The level and temperature in the second subsystem are affected by those in the first subsystem while the controller outputs in the first subsystem are affected by the controlled variables in the second subsystem.

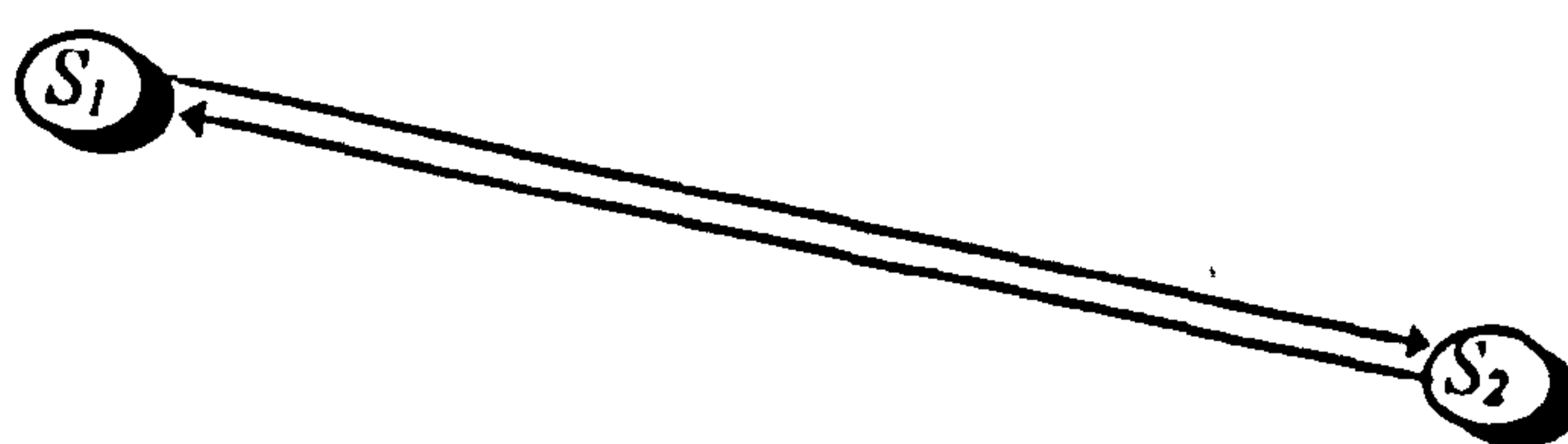


Figure 9.2 - The mixing process directed graph



Therefore, according to equation (9.1) the Connection Matrix for the mixing process is given by the following equation,

$$C = \begin{matrix} & S_1 & S_2 \\ \begin{matrix} S_1 \\ S_2 \end{matrix} & \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \end{matrix} \quad (9.6)$$

Moreover, the on-line information in the first subsystem includes level, temperature, cold water input flow and hot water input flow measurements. Therefore, according to equation (9.4) the Self-Causal Matrix for the first subsystem is represented by equation (9.7),

$$CS_1 = \begin{matrix} & L_1 & T_1 & Q_c & Q_h \\ \begin{matrix} L_1 \\ T_1 \\ Q_c \\ Q_h \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \end{matrix} \quad (9.7)$$

where the following notation is used for labels on the top and left of the matrix,

- $L_1$  - is the level measurement in tank 1;
- $T_1$  - is the temperature measurement in tank 1;
- $Q_c$  - is the cold water input flow;
- $Q_h$  - is the hot water input flow.

In the mixing process, either  $Q_c$  or  $Q_h$  can affect both level and temperature in tank 1, respectively  $L_1$  and  $T_1$ . However, according the rule based controller implemented for the mixing process and described in chapter 4., the cold water input flow,  $Q_c$ , is used to control level and the hot water input flow,  $Q_h$ , is used to control temperature. Consequently, the effect of  $Q_c$  on  $T_1$  and the effect of  $Q_h$  on  $L_1$  can be eliminated by the feedback control loops. Therefore, due to the decoupling scheme implemented in the mixing process control, equation (9.7) indicates that the cold water input flow,  $Q_c$ , only can affect itself and the level in tank 1,  $L_1$ , while the hot water input flow,  $Q_h$ , only can affect itself and temperature in tank 1,  $T_1$ .

The on-line information about the second subsystem is the level and temperature in tank 2, respectively  $L_2$  and  $T_2$ . Then, the Self-Causal Matrix for the second subsystem is given by the following expression,

$$CS_2 = \begin{matrix} & L_2 & T_2 \\ \begin{matrix} L_2 \\ T_2 \end{matrix} & \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{matrix} \quad (9.8)$$

where the labels on the top and the left of the matrix,  $L_2$  and  $T_2$ , have respectively the following meaning,

$L_2$  - level measurement in tank 2;

$T_2$  - temperature measurement in tank 2.

Because in the second subsystem each measurement variable can only affect itself, according to equation (9.4) only the diagonal elements of the Self-Causal Matrix, which is represented by equation (9.8), take the value 1.

Since  $Q_c$ ,  $Q_h$ ,  $L_1$  and  $T_1$  belong to subsystem  $S_1$  while  $L_2$  and  $T_2$ , belong to subsystem  $S_2$ , according to equation (9.3) the Measurement Causal Matrix from subsystem  $S_1$  to subsystem  $S_2$  can be expressed by equation (9.9).

$$CM_{12} = \begin{matrix} & L_2 & T_2 \\ \begin{matrix} L_1 \\ T_1 \\ Q_c \\ Q_h \end{matrix} & \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \end{matrix} \quad (9.9)$$

As equation (9.9) states, the controlled variables cold and hot water input flows, respectively  $Q_c$  and  $Q_h$ , cannot directly affect the measurement variables level and temperature in tank 2, respectively  $L_2$  and  $T_2$ , since their influence on these variables is exerted through the measured variables level and temperature in tank 1,  $L_1$  and  $T_1$  respectively. Therefore, the level in tank 2 is only directly affected by the measurement variable level in tank 2, while the measurement variable temperature in tank 2 is only directly affected by the measurement variable temperature in tank 1.

The Measurement Causal Matrix from subsystem  $S_2$  to subsystem  $S_1$ , is given by equation (9.10),

$$CM_{21} = \begin{matrix} & L_1 & T_1 & Q_c & Q_h \\ \begin{matrix} L_2 \\ T_2 \end{matrix} & \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{matrix} \quad (9.10)$$



The level and temperature in tank 2 can affect both cold and hot water controller outputs. However, the Measurement Causal Matrix from subsystem  $S_2$  to subsystem  $S_1$ , indicates that  $L_2$  can only affect  $Q_c$ , and  $T_2$  can only affect  $Q_h$ , as is expressed in equation (9.10). This is due to the fact quoted above, that a decoupling scheme to eliminate the interactions between control loops has been implemented and, therefore, cold water input flow,  $Q_c$ , and hot water input flow,  $Q_h$ , are dominantly affected by level in tank 2,  $L_2$ , and temperature in tank 2,  $T_2$ , respectively.

Based on the above described process structures, fault detection heuristic rules can be developed. We start with the process controlled variables, which are level and temperature in tank 2. Hence, consider the hypothetical situation where  $L_2$  presents an abnormal behaviour taking a value lower than its setpoint. The procedure for generating the fault detection heuristic rules is based on a search to find if there are any measured variables in the subsystem  $S_2$  (where  $L_2$  belongs) which can affect  $L_2$ . From equation (9.8), it can be seen that no such variables exist. However, equation (9.6) indicates that subsystem  $S_1$  can affect subsystem  $S_2$  and, furthermore, equation (9.9) shows that only  $L_1$  in subsystem  $S_1$  can affect  $L_2$ . Then  $L_1$  should be examined and, if  $L_1$  is decreasing  $S_1$  will be the subsystem responsible for the abnormal behaviour in  $L_2$ . Under this assumption, from equation (9.7) it can be seen that only the cold water input flow,  $Q_c$ , can affect  $L_1$ . Then, if the cold water input flow,  $Q_c$ , is decreasing the search procedure is terminated since both subsystems have been explored and the following heuristic fault detection rule should be generated,

***IF* ( $L_2$  is lower than its setpoint **AND**  
 $L_1$  is decreasing **AND**  
 $Q_c$  is continuously decreasing) **THEN**  
enable fault detection flag**

However, if the measurement variable  $L_1$  is not decreasing, subsystem  $S_2$  will be responsible for the abnormal behaviour of the measurement variable level in tank 2. Equation (9.8) suggest that only  $L_2$  can affect itself in subsystem  $S_2$ . In such a situation the following rule is generated,

***IF* ( $L_2$  is lower than its setpoint **AND**  
 $L_1$  is not decreasing **AND**  
 $L_2$  is continuously decreasing) **THEN**  
enable fault detection flag**



The notion of "continuously decreasing" has been introduced, in order to avoid false fault detection situations due to small perturbations in the process or even noise problems in a real implementation of the fault detection system. Moreover, the fault detection heuristic rules, for the situation where the level in tank 2,  $L_2$ , is higher than its setpoint, are generated in a similar way as above for the opposite situation.

For the other controlled variable, which is temperature in tank 2,  $T_2$ , there are also two situations that could indicate an abnormal process behaviour. As quoted above for level, one is that  $T_2$  is lower than its setpoint and, another one is that  $T_2$  is higher than its setpoint. Hence, for instance let us consider the situation where  $T_2$  is higher than its setpoint. Equation (9.8) suggest there is no variable in subsystem  $S_2$  which can affect  $T_2$ , while equation (9.6) suggest that subsystem  $S_1$  can affect subsystem  $S_2$ . Moreover, equation (9.9) shows that only  $T_1$  in the first subsystem can affect  $T_2$  in the second subsystem. Then, if  $T_1$  is increasing, subsystem  $S_1$  will be responsible for the abnormal behaviour in  $T_2$ . Under this assumption, from equation (9.7) we can see that only the hot water input flow,  $Q_h$ , can affect the temperature in tank 1,  $T_1$ , and, then if  $Q_h$  is increasing the search procedure is terminated since both subsystems have been explored and the following heuristic rule is generated,

***IF*** ( $T_2$  is higher than its setpoint ***AND***  
 $T_1$  is increasing ***AND***  
 $Q_h$  is continuously increasing) ***THEN***  
**enable fault detection flag**

The list of single faults that can occur in the mixing process simulation is presented in chapter 6., while the list of double simultaneous faults considered for the mixing process is presented in chapter 7.. It can be seen that we do not consider sensor failures. Hence, temperature in tank 2 should follow temperature in tank 1 apart from a small lag and, hence, the procedure for generating fault detection heuristic rules, under the situation quoted above, is terminated. The fault detection heuristic rules, for the situation where  $T_2$  is lower than its setpoint, are generated in a similar way as above.

In order to cope with valve saturation situations, the deep knowledge quoted above has been combined with shallow knowledge to implement the knowledge based fault detection system. The performance of the fault detection system applied on the simulation studies of the mixing process, is discussed in chapter 10.. A combined approach, the knowledge based system and a fuzzy neural network, has been implemented and successfully applied for fault detection and diagnosis on the mixing process.



## 9.5 Formulation of Fault Detection Rules for the CSTR

A detailed description of the continuous stirred tank reactor (CSTR), as well as the development of its dynamic model and its qualitative model, have already been given in chapter 5.. Control of some process variables, such as temperature and level in the reactor, as well as the recycle flow rate, which are controlled by feedback control systems (cascade control for the case of temperature), is considered.

Following a similar procedure as described above and already applied to the mixing process, the CSTR process is decomposed into three subsystems. The first subsystem,  $S_1$ , consists of the following components: external feed reactant elements, which includes pipe 1 and associated sensors. The second subsystem,  $S_2$ , includes the following components: reaction vessel, pipe 2 pump, pipe 3, valve 1, pipe 11 and associated sensors. The remaining components form the third subsystem,  $S_3$ , which are all the components associated with the heat exchange part of the process. The Directed Graph corresponding to this decomposition is shown in Figure 9.3, from which it can be seen that the first subsystem can affect the second subsystem while the second subsystem cannot affect the first one, and both subsystems,  $S_2$  and  $S_3$  can affect each other.

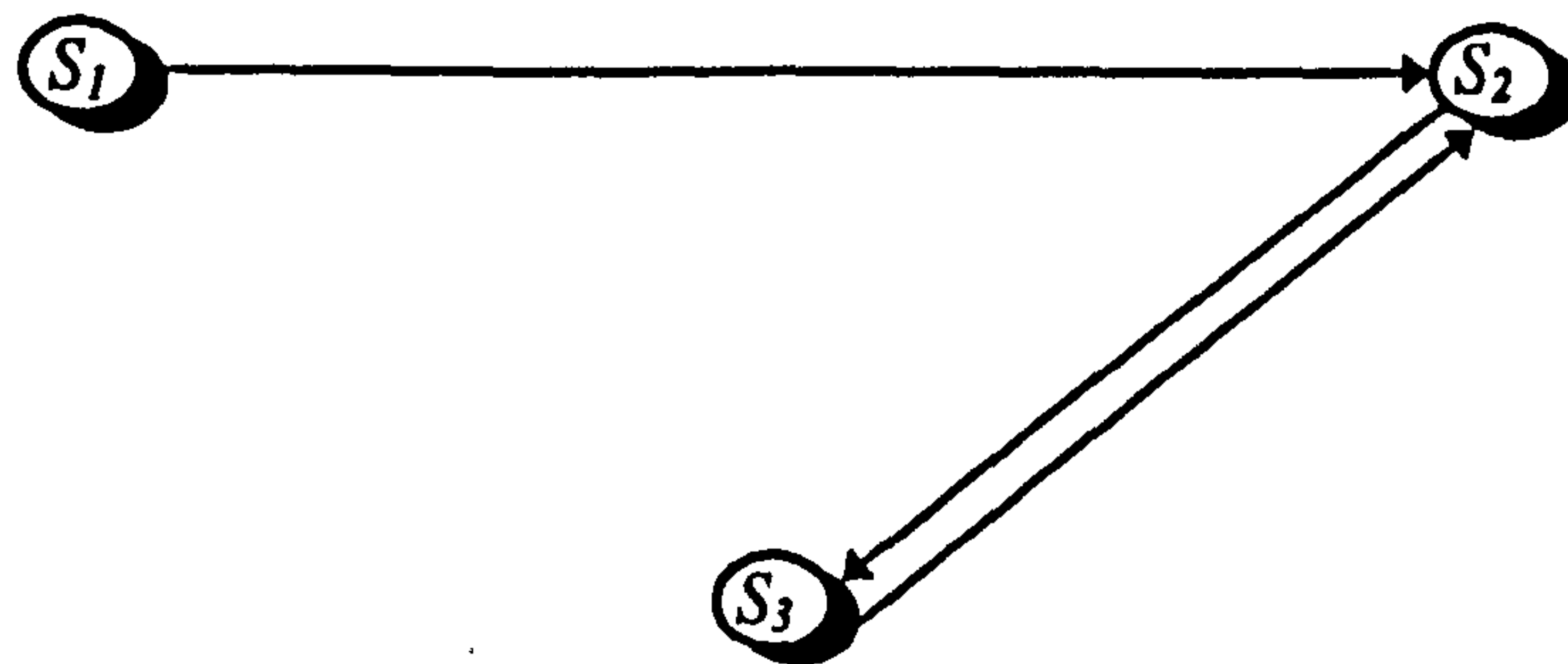


Figure 9.3 - Continuous stirred tank reactor directed graph

Thus, in a similar way as quoted above for the mixing process, and according to equation (9.1), the Connection Matrix for the continuous stirred tank reactor is given by the following equation,

$$C = \begin{matrix} & \begin{matrix} S_1 & S_2 & S_3 \end{matrix} \\ \begin{matrix} S_1 \\ S_2 \\ S_3 \end{matrix} & \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \end{matrix} \quad (9.11)$$

To perform the process behaviour analysis ten measurement variables have been considered. For the first subsystem,  $S_1$ , three measurements  $Q_1$ ,  $T_1$  and  $C_{ao}$ , are considered which

are the flow rate, temperature, and concentration of the external feed reactant respectively. Hence, according to equation (9.4) the Self-Causal Matrix for the first subsystem is given by the following equation:

$$CS_1 = \begin{matrix} & Q_1 & T_1 & C_{a_0} \\ \begin{matrix} Q_1 \\ T_1 \\ C_{a_0} \end{matrix} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{matrix} \quad (9.12)$$

which suggests that the three measurement variables considered for the first subsystem cannot affect each other.

Following an analogous procedure, the Self-Causal Matrix for the second subsystem,  $S_2$ , is given by equation (9.13):

$$CS_2 = \begin{matrix} & L & T & Q_4 & C_a \\ \begin{matrix} L \\ T \\ Q_4 \\ C_a \end{matrix} & \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{matrix} \quad (9.13)$$

where the following notation is used,

$L$  - stands for level in the reactor;

$T$  - stands for temperature in the reactor;

$Q_4$  - stands for flow rate through valve 1;

$C_a$  - stands for concentration of the reactant in the product.

For the third subsystem of the CSTR plant,  $S_3$ , the following measurement variables are considered,

$Q_2$  - flow rate through valve 3;

$Q_5$  - flow rate through valve 2;

$T_5$  - temperature of cold water entering heat exchanger.

and, hence, the Self-Causal Matrix for the third subsystem is given by the following equation:



$$CS_3 = \begin{matrix} & Q_2 & Q_5 & T_5 \\ \begin{matrix} Q_2 \\ Q_5 \\ T_5 \end{matrix} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{matrix} \quad (9.14)$$

Moreover, according to equation (9.3) the Measurement Causal Matrix from subsystem  $S_1$  to subsystem  $S_2$ , is given by the following equation:

$$CM_{12} = \begin{matrix} & L & T & Q_4 & C_a \\ \begin{matrix} Q_1 \\ T_1 \\ C_a \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \end{matrix} \quad (9.15)$$

From equation (9.15), it can be seen that the measurement variables flow rate and temperature of external feed reactant can directly affect the controlled variables level and temperature in the reactor vessel respectively, and the external feed reactant concentration can directly affect both the temperature in the reactor and the reactant concentration in the product.

The Measurement Causal Matrix from subsystem  $S_2$  to subsystem  $S_3$  is:

$$CM_{23} = \begin{matrix} & Q_2 & Q_5 & T_5 \\ \begin{matrix} L \\ T \\ Q_4 \\ C_a \end{matrix} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{matrix} \quad (9.16)$$

Equation (9.16) shows that the process variables level in the reactor and flow rate through control valve 1, which are measurement variables of the second subsystem, can only directly affect the flow rate through valve 3 in the third subsystem.

The Measurement Causal Matrix from subsystem  $S_3$  to subsystem  $S_2$ , is given by the following equation:

$$CM_{32} = \begin{matrix} & L & T & Q_4 & C_a \\ \begin{matrix} Q_2 \\ Q_5 \\ T_5 \end{matrix} & \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \end{matrix} \quad (9.17)$$

Equation (9.17) suggests that the flow rate through valve 3, which belongs to the third subsystem, can directly affect both level in the reactor and flow rate through valve 1, in the second subsystem. Moreover, both the flow rate through valve 2 and the temperature of cold water entering the heat exchanger, which are measurement variables of subsystem  $S_3$ , can only directly affect the temperature into the reactor, which is a measurement variable of subsystem  $S_2$ .

Abnormal behaviour detection is similar to that for the mixing process. Fault detection heuristic rules are generated from the knowledge on system structures and component functions in a similar way as quoted above for the mixing process. Considering controlled variables abnormal behaviour, the reasoning behind the generation of the heuristics is conducted to causally search any measured variable or variables which could be responsible for the observed abnormality. These variables could be in the same subsystem where the abnormality is observed or in any other subsystem. Hence, for instance, let us consider the situation where the level in the reactor is presenting an abnormal behaviour. In such a situation two possibilities must be considered, one is that the level is higher than its setpoint and another is that the level is lower than its setpoint. Consider the first possibility. From equation (9.13) it can be seen that only  $Q_4$  in  $S_2$  can affect the level in the reactor,  $L$ . Therefore, if  $Q_4$  is continuously decreasing then it is responsible for  $L$  being higher than its setpoint and the following fault detection rule is generated,

***IF* ( $L$  is higher than its setpoint *AND*  
 $Q_4$  is continuously decreasing) *THEN*  
enable fault detection flag**

However, if  $Q_4$  is not responsible for the  $L$  abnormal behaviour, then equation (9.15) suggests that  $Q_1$  in the first subsystem can affect  $L$  in the second subsystem. Therefore, if  $Q_1$  is continuously increasing then it is responsible for the observed abnormal behaviour in  $L$ , and the following heuristic rule is achieved,

***IF* ( $L$  is higher than its setpoint *AND*  
 $Q_4$  is not decreasing *AND*  
 $Q_1$  is continuously increasing) *THEN*  
enable fault detection flag**

But if  $Q_1$  is not responsible for the  $L$  abnormal behaviour, then equation (9.17) shows that only  $Q_2$  in the third subsystem can directly affect  $L$  in the second subsystem. Hence, if  $Q_2$  is continuously increasing then it is responsible for  $L$  being higher and the following rule is achieved,



***IF* (*L* is higher than its setpoint ***AND***  
 $Q_4$  is not decreasing ***AND***  
 $Q_1$  is not increasing ***AND***  
 $Q_2$  is continuously increasing) ***THEN***  
enable fault detection flag**

The formulation of fault detection rules for the situation where the level in the reactor vessel is lower than its setpoint is similar to the above procedure, as well as for the other controlled process variables.

In the developed fault detection system, nineteen such heuristic rules have been used in the simulation studies of the CSTR process. Moreover, to cope with valve saturation effects and avoid false fault detection situations when more than one setpoint change is performed, some shallow knowledge, based on the operational process experience, has been used to build up the knowledge base. As presented in the next chapter, the knowledge based fault detection system joined together with a fuzzy neural network, which is responsible for the fault diagnosis task, has been applied successfully for fault detection and diagnosis of single and multiple abrupt faults and incipient faults on simulation studies of the continuous stirred tank reactor.

## 9.6 Conclusions

A systematic methodology for formulating fault detection heuristic rules from knowledge of system structure and component functions has been developed. Therefore, a method to describe the system structures is presented in sub chapter 9.2. The reasoning behind the development of fault detection heuristic rules is described in sub chapter 9.3. The application of such a methodology to a mixing process, as well as to a continuous stirred tank reactor, respectively, is presented in sub chapters 9.4 and 9.5.

Since structural decomposition corresponds to plant topology, such a method could be easier to implement. Advantages of a rule based format are that rules are efficient to evaluate and the heuristic rules based on deep knowledge can be combined with other rules based on shallow knowledge. Moreover, the apparent advantages of the proposed method for eliciting fault detection heuristic rules based on deep knowledge of the process under consideration are as follows:

1. Given a plant description, the program designer is able to shortcut the laborious process of eliciting empirical associations from a human expert;

2. The reasoning method employed is device independent, so it is not necessary to tailor the inference machinery for different applications;
3. The methodology is systematic, so may be appropriate for large scale processes which consist of a large number of components as well as strong interactions between them.

The successful application of this method for developing fault detection heuristic rules for the simulated mixing process and for a simulated continuous stirred tank reactor suggest that the method provides a systematic and efficient approach for the design of on-line rule based fault detection systems. Moreover, since the fault detection task is not dependent on threshold values the performance of an overall fault detection and diagnosis system could be increased.



## **Chapter 10**

# **On-Line Fault Detection and Diagnosis Using a Knowledge Based System Coupled with a Fuzzy Neural Network**

### **10.1 Introduction**

Fault detection and diagnosis systems based on conventional techniques are usually supported by linear system models. For non-linear processes, the traditional approach is to linearize the system model around the system operating point. This approach is effective for many non-linear processes if the operating range is limited and the fault detection and diagnosis system has been designed to be robust enough to tolerate small perturbations around the operating point. However, for systems with high nonlinearity and a wide dynamic operating range, the linearized approach fails to give satisfactory results. One solution is to use a large number of linearized models corresponding to a range of operating points which is not very practical for real-time applications (Chen 1995). Hence, in this chapter a fault diagnosis approach based on artificial neural networks is presented. Because artificial neural networks can be trained to have the required relationships between inputs and outputs, they can be used to overcome difficulties in conventional techniques for dealing with nonlinearity. Neural networks are properly aimed at processes that are ill-defined, complex, non-linear and stochastic.

The use of artificial neural networks for fault detection and diagnosis purposes has received increasing attention in both research and application. The number of publications about this subject have demonstrated the promise of this new tool (Watanabe *et al.* 1989 and 1994,



Venkatasubramanian and Chan 1989, Naidu *et al.* 1990, Himmelblau *et al.* 1991, Hoskins *et al.* 1991, Sorsa *et al.* 1991, Willis *et al.* 1991, Zhang and Roberts 1992a, Sorsa and Koivo 1993, Kavuri and Venkatasubramanian 1994, Patton *et al.* 1994, Zhang and Morris 1994). In the present approach, a fuzzy neural network is used, which combines the capability of fuzzy reasoning in handling uncertain information and the capability of neural networks in learning from examples. In contrast with a conventional feed forward neural network, this fuzzy network has an additional fuzzification layer which converts the increment in each on-line measurement into seven fuzzy sets: negative large (*nlarge*), negative medium (*nmedium*), negative small (*nsmall*), zero, positive small (*psmall*), positive medium (*pmedium*) and positive large (*plarge*). By using these fuzzy sets to describe the process variables real behaviour, due to the fact that similar training patterns are transformed into the same fault symptoms, training data will be compressed and training effort can be eased. Moreover, the fuzzy approach also makes the system less sensitive to measurement noise (Zhang and Morris 1994).

In order to achieve on-line fault detection and diagnosis in the presence of transient behaviours, the system dynamics have to be considered. However, most publications only deal with processes under steady-state conditions. In these applications, neural networks were used to examine the possible fault or faults in the process under concern and give a fault classification signal to declare whether or not the process is faulty. This procedure may be suitable for diagnosing faults for some processes under steady state conditions, but this is not the case for diagnosing faults in dynamic processes because the change in the neural network inputs can also affect certain features of the neural network outputs. Following this approach, the fault detection and diagnosis system could give incorrect information about a fault or faults in the process when in the presence of transient behaviours. In order to overcome this problem, a hybrid fault detection and diagnosis system, which is a combined approach of the deep/shallow knowledge based system, described in the last chapter, with a fuzzy neural network, is considered in this chapter. The knowledge based system is responsible for detecting a fault in the process under concern while the fuzzy neural network locates the hypothetical fault or faults.

This chapter is organised as follow. Sub chapter 10.2 presents the fault detection and diagnosis system architecture. In sub chapter 10.3 a general description of classical artificial neural network is presented. Sub chapter 10.4 discusses the advantages of using a fuzzy neural network for fault diagnosis purposes instead a classical neural network, as well as giving a fuzzy neural network topology. In sub chapter 10.5 an extension of the classical backpropagation learning algorithm is proposed for performing the learning task. In sub chapters 10.6 and 10.7 the fault detection and diagnosis system is applied to the mixing process and to the continuous stirred tank reactor, respectively, which are presented in previous chapters; in these simulation studies single



and multiple simultaneous abrupt faults and incipient faults have been considered. The performance of the fault detection and diagnosis system is analysed in sub chapter 10.8. Sub chapter 10.9 provides some concluding remarks.

## 10.2 System Architecture

When neural networks were revived in recent years they were labelled by some as sixth-generation computing (Turban 1992). This labelling gave the erroneous impression that the fifth-generation computing, of which knowledge based systems are a major part, is going to be replaced. As a matter of fact, while in some cases neural networks can perform tasks better or faster than knowledge based systems, in most instances the two technologies are not in competition. Furthermore, the characteristics of both techniques are so different that they can complement each other in some practical applications.

Knowledge based systems perform reasoning using pre-established heuristic rules for a well defined and narrow domain. They combine knowledge bases of rules and domain specific facts with information about specific instances of problems, provided by a domain expert. Ideally, reasoning can be explained and the knowledge bases can be easily modified or updated, independently of the inference engine, as new rules become known.

A major limitation of the knowledge based approach arises from the fact that domain experts do not always think in terms of rules. Moreover, domain experts may not be able to explain their line of reasoning, or they may explain it incorrectly. Thus, in many cases, it is difficult or even impossible to build the necessary knowledge base. In order to overcome this, or other limitations, neural networks could be used.

As quoted above, the neural network approach relies on training data to model the system. Particular applications are developed by establishing an appropriate training set that allows the system to learn and generalise for operation on future input data. Inputs that match the training data exactly are recognised and identified, while new data or incomplete and noise versions of the training data can be matched closely to patterns recognised by the system (Zhang and Roberts 1992a).

For well behaved systems, with well defined rules, knowledge based systems can be developed to provide good performance. In contrast to neural network approaches, knowledge is represented as numeric weights and, hence, the rules and the reasoning process are not readily explainable. Neural networks can be preferable to knowledge based systems when rules are not known either because the topic is too complex or no domain expert is available. If training data can



be generated the system may be able to learn enough information to perform as well as, or better than, a knowledge based system. This approach also has the benefit of easy maintenance, since modifications are achieved by retraining the neural network with an updated data set, avoiding programming changes and rule reconstruction. Moreover, as described below, the use of a special type of neural network, a so called fuzzy neural network, could be more suitable for fault diagnosis purposes. Such a neural network combines the capability of fuzzy reasoning in handling uncertain information with the capability of neural networks in learning from examples.

From the above considerations, there is motivation to implement a hybrid fault detection and diagnosis system, which consists of a knowledge based approach for fault detection combined with a fuzzy neural network approach for fault diagnosis. The overall system can be seen as a distributed intelligent system in the sense that both subsystems are capable of functioning independently. They can interface with each other through communication lines, that is by transfer of data. Once a fault has been detected by the knowledge based system, the fault diagnosis system is triggered to locate the hypothetical fault or faults in the process under concern. On-line measurement data is fed forward through the fuzzy neural network and the corresponding output values analysed.

As in the computational systems presented in previous chapters, the overall computational system presented in this chapter still can be seen as a two layer configuration. In the lower layer the process under consideration is simulated through its dynamic model and control of some process variables is performed. It is also through the lower layer that a fault, or faults in the process under concern, can be simulated. The upper layer has supervisory functions with the main aim of detecting and diagnosing faults introduced in the lower layer. Figure 10.1 depicts the overall computational system, which has been implemented through a TURBO C++ program.

The information handled by the fault detection and diagnosis system is basically the changes which occur in on-line measurement variables. However, for simplicity, all the changes observed in the measurement variables used by the fault detection and diagnosis system have been normalised into the range  $[-1, +1]$ . From Figure 10.1, it can be seen that the normalisation task is performed by a "Data Normalisation" module. This module interfaces the lower layer with the upper layer. On-line measurement variables are sampled with a pre-defined sampling time and their values are retained. With two consecutive samples retained, the changes in the measurement variables which occur between sampling times are evaluated and then converted into the normalised range  $[-1, +1]$ . To perform this task, it is necessary to define a parameter for each variable which makes the normalisation of the different process variable changes.

In previous implemented fault detection and diagnosis, described in previous chapters, normalisation parameters have also been used. During the studies conducted to develop such



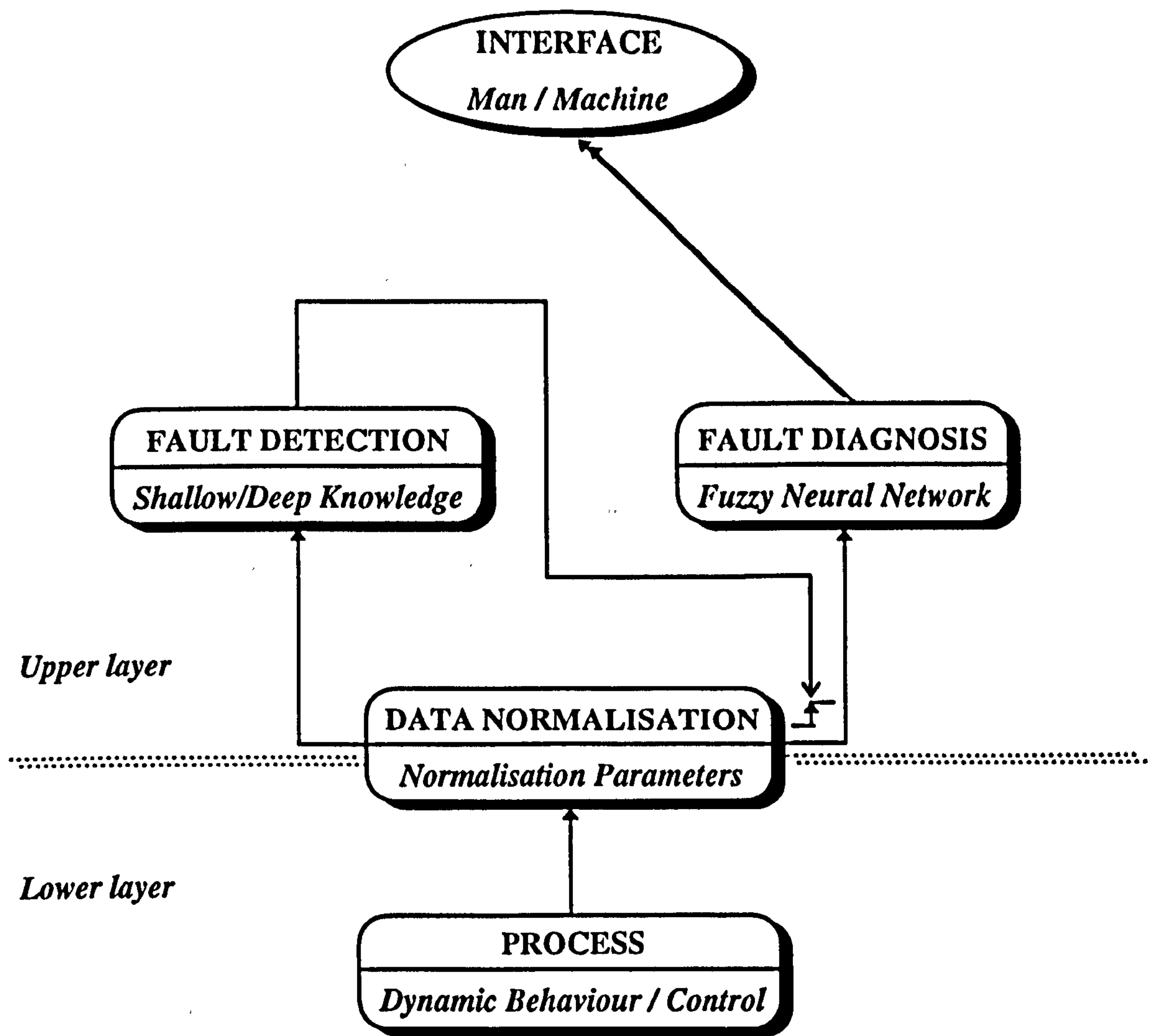


Figure 10.1 - Architecture of a on-line fault detection and diagnosis system

systems, it is found that the normalisation parameters could affect the performance of the overall system and, hence, should be set properly. In contrast, since in the current approach the fault detection task is not based on threshold values, but is performed through a knowledge based system and the fault diagnosis is achieved through a fuzzy neural network previously trained with data acquired with the normalisation parameters already defined, then the performance and reliability of the overall fault detection and diagnosis system is not affected by such normalisation parameters.

At each sampling time the normalised data is passed to a "Fault Detection" module. This module is basically a knowledge based system consisting of deep and shallow knowledge of the process under concern. The fault detection heuristic rules have been generated following the procedure described in the last chapter. From the on-line information transferred to the knowledge based system, a forward chaining inference engine tries to match the heuristic rules pre-conditions against the current state. If at least one rule is fired a fault detection flag is enabled and the fault diagnosis module is triggered. Then, the fault diagnosis task is performed through a fuzzy neural network with the topology described in sub chapter 10.4. Therefore, if the fault diagnosis module

has been triggered, the normalised changes observed in the measurement variables are fed forward through the fuzzy neural network. Thus, a particular diagnosis could be achieved if any fuzzy neural network output has a value close to one. In the current approach it is considered that any fuzzy neural network output with a value greater than 0.8 suggests that the corresponding hypothetical fault has occurred in the process under consideration.

After a fault or faults have been diagnosed the computational system keeps the operator informed about that fact through the "Interface Man/Machine" module. This module provides to the operator the fault or faults location, as well as the values of the neural network outputs which have been obtained during the diagnosis phase. Moreover, since we are using simulation studies, it is possible to know the time instant when the fault or faults have been initiated and, therefore, the fault detection and diagnosis time is also provided through the man/machine interface. This parameter allows us to evaluate the performance of the overall fault detection and diagnosis system.

Since the current research work has been carried out through simulation studies the man/machine interface module has been also used for allowing the simulation of faults in the process under consideration. Therefore, if a faults simulation is chosen from the main screen on the application an iterative menu appears as shown in Figure 10.2. From this menu the user should

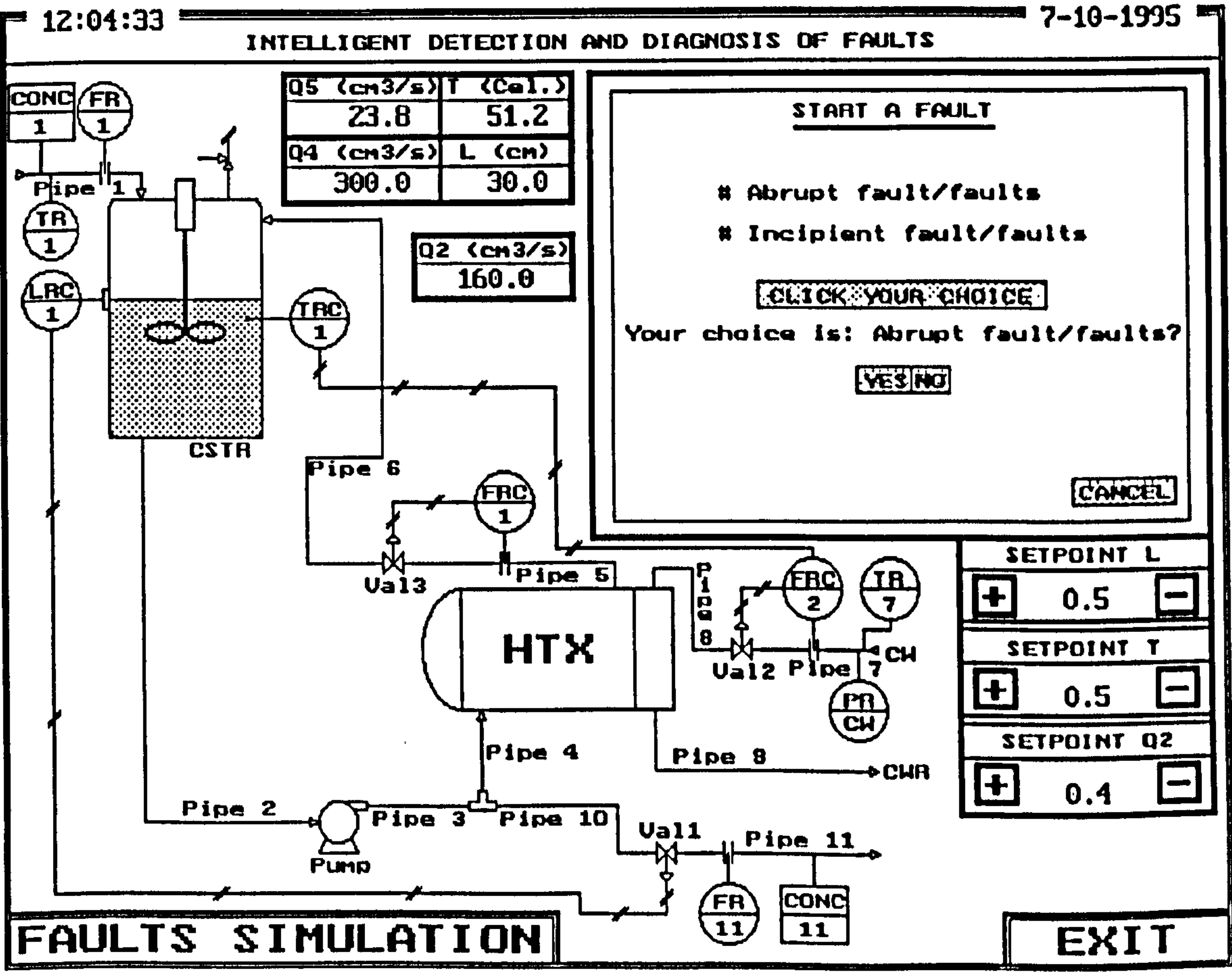


Figure 10.2 - Faults simulation main menu



chose between abrupt fault/faults and incipient fault/faults. A similar menu is used for defining if the user wants to simulate single or double multiple simultaneous faults. At this stage, further similar menus are used to initiate a specific fault or faults simulation. However, if at the first menu the user has chosen incipient fault/faults simulation, before the fault or faults will be initiated the user is requested to provide the speed of the fault or faults development through the menu shown in Figure 10.3. The examples depicted in Figures 10.2 and 10.3 are for the simulation of the continuous stirred tank reactor but they are process independent. This means that the same interface is used in simulation studies conducted with the mixing process.

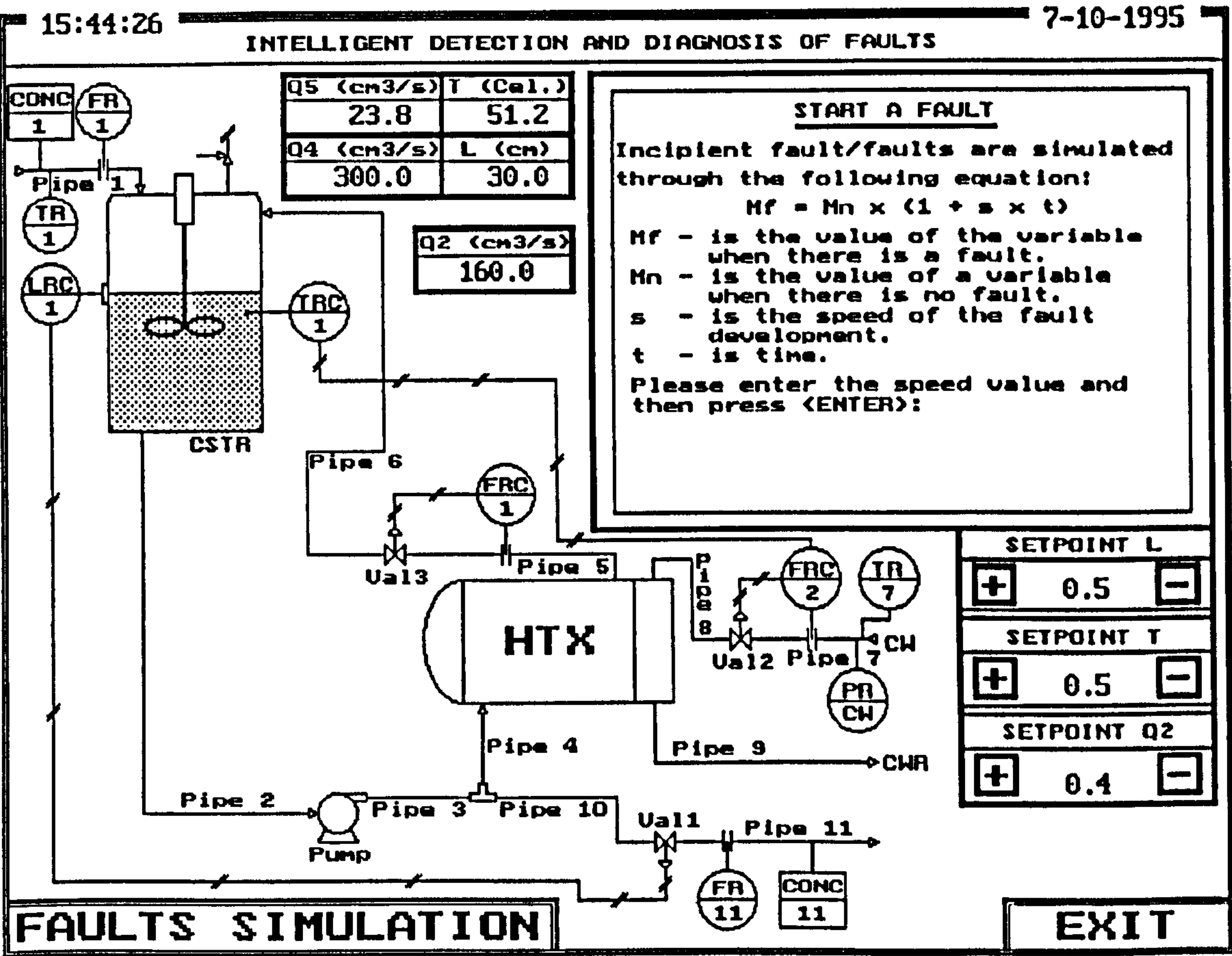


Figure 10.3 - Man/machine interface for incipient fault simulation

### 10.3 Artificial Neural Networks

This sub chapter is an introduction to Artificial Neural Networks with the aim for providing the background for the Fuzzy Neural Networks field which is the subject of the next sub chapter. Artificial neural networks try to model the brain's cognitive process. In contrast with conventional single-processor computers, the brain is considered to have a multiprocessor



architecture that is highly interconnected. This architecture can be, and has been, described as parallel distributed processing. In this way, artificial neural networks are also referred to in the literature as neurocomputers, connectionist networks, parallel distributed processors, etc. Throughout this chapter the author uses simply the term "neural networks".

Neural networks originated as a model of how the brain works and have a long history. Indeed, this research has its beginnings in psychology. The theories of Freud, and other nineteenth-century psychologists, laid the groundwork of ideas that was to give birth to early neural networks research (Blum 1992). In a neural network, the unit analogous to the biological neurone is referred in neural networks literature as a "processing element" or a "processing unit". In this chapter the term processing element is adopted.

Development of detailed mathematical models began more than fifty years ago (McCulloch and Pitts 1943, Hebb 1949). More recent work has led to a new resurgence of the field (Hopfield 1982, Hopfield and Tank 1986, Rumelhart and McClelland 1986). This new interest is due to the development of new neural network topologies and algorithms, and new analogue very large scale integration techniques. All these models attempt to achieve good performance via dense interconnection of simple computational elements. In this respect, neural networks structure is based on our understanding of the biological nervous system. Recent interest is also driven by the realisation that human-like performance in the areas of speech and image recognition will require enormous amounts of processing. Neural networks provide one technique for obtaining the required processing capacity using large numbers of simple processing elements operating in parallel (Lippmann 1987).

Instead of performing a program of instructions sequentially, neural network models explore many competing hypotheses simultaneously using massively parallel networks composed of many computational elements connected by links with variable weights. Unlike traditional expert systems, where knowledge is made explicit in the form of rules, neural networks generate their own rules by learning from being shown examples. Learning is achieved through a learning rule which adapts or changes the connection weights of the network in response to the example inputs and, in some situations, the desired outputs of those inputs. Moreover, whereas traditional computing systems are rendered useless by even a small amount of damage to memory, neural networks based computing systems are fault tolerant. Fault tolerance refers to the fact that in most neural networks, if some processing elements are destroyed, disabled, or their connections altered slightly, then the behaviour of the network as a whole is only slightly degraded. As yet more processing elements are destroyed the behaviour of the neural network is degraded just a bit further. Performance suffers, but the system doesn't come to an abrupt halt. Neural network based computing systems are fault tolerant because information is not contained in one place, but is distributed throughout the system.



This characteristic of graceful degradation makes these systems extremely well suited for applications where failure means disaster.

Typically, a neural network is presented with a training set consisting of a group of examples from which the neural network can learn. These examples, known as training patterns, are represented as vectors, and can be taken from such sources as images, speech signals, sensor data, robotic arm movements, financial data, and diagnosis information (Dayhoff 1990). In diagnostics, a pattern (a set of measurements or symptoms) acts as antecedents from which we can infer a classification (diagnosis) of each pattern. The neural networks training procedure is described, in detail, below.

Like brains, neural networks recognise patterns we cannot even define. We call this property recognition without definition. Recognition without definition characterises much intelligent behaviour in that it enables systems to generalise (Kosko 1992). Since a neural network has the ability to generalise on the tasks for which it is trained, fault diagnosis seems to be a promising field for their application. From this point of view, the ability to generalise may enable the neural network to provide the correct answer when presented with a new input pattern that is different from the inputs in the training set. This means that the neural network has the capability to provide a correct diagnosis under a new faulty scenario. However, during the present research work, it has been observed that to achieve an effective generalisation behaviour from a neural network, the training session must be limited in iterations, so that no "overlearning" takes place, and the training set must include a variety of examples that are a good preparation for the generalisation task.

An example of a typical processing element for a neural network is depicted in Figure 10.4. On the left are the multiple inputs to the processing element, each arriving from another element, which is connected to the element shown at the centre. Each interconnection has an associated strength, which is called the connection weight, given as  $w_{j1}$ ,  $w_{j2}$ , .....,  $w_{jn}$ . Furthermore, the processing element performs a weighted sum on the inputs and uses a transfer function,  $f$ , to compute its output. This transfer function can be a threshold function, which only passes information if the weighted sum reaches a certain value, or it can be a continuous function of the weighted sum. The output value of the transfer function is generally passed directly to the output path of the processing element.

Moreover, the output path of a processing element can be connected to input paths of other processing elements through connection weights. Since each connection has a corresponding weight, the signals on the input lines to a processing element are modified by these weights prior to being summed. In itself, this simplified model of a processing element is not very interesting; the interesting properties result from the ways processing elements are interconnected.



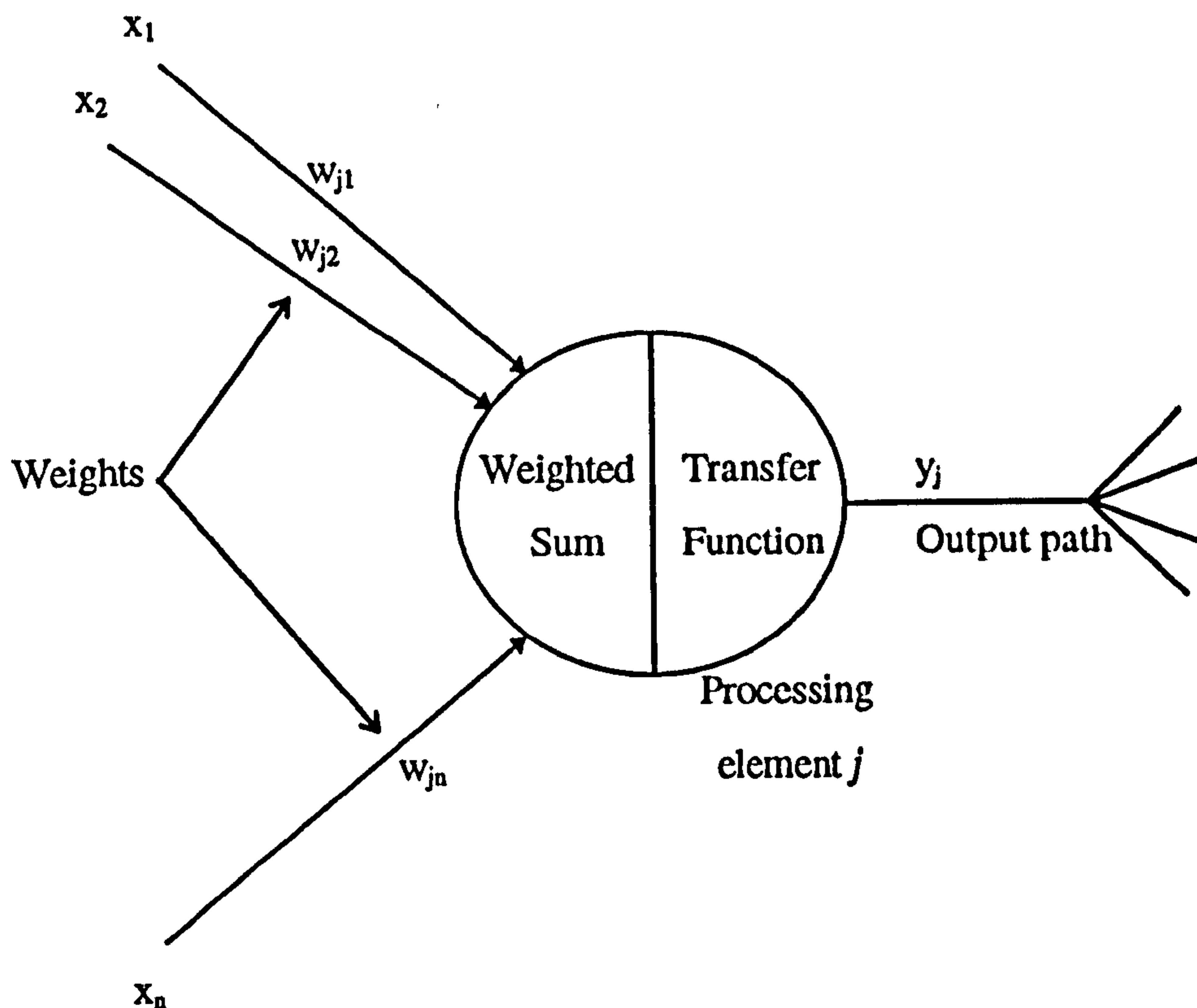


Figure 10.4 - Basic processing element

Artificial neural networks consist of numerous, simple processing elements that we can globally program for computation. We can program or train neural networks to store, recognise and associatively retrieve patterns or database entries; to solve combinatorial optimisation problems; to filter noise from measurement data; to control ill-defined problems; in summary, to estimate sampled functions when we do not know the form of the functions (Kosko 1992). Artificial neural networks have been shown to possess a good approximation capability for a wide range of non-linear functions (Hornik *et al.* 1989, Park and Sandberg 1991, Wang *et al.* 1992).

A neural network consists of many processing elements joined together in the above manner. Processing elements are usually organised in groups which are called layers. A typical multilayer artificial neural network consists of a sequence of layers with full or random connections between successive layers, as shown in Figure 10.5. Each layer of the neural network consists of computing nodes. There are usually two layers with connections to the outside world, which are an input buffer where data is presented to the neural network, and an output buffer responsible to hold the response of the neural network to a given input. Layers between the input and output buffers are called hidden layers.

There are two main phases in the operation of a neural network which are called "Learning" and "Recall". Learning is the process of adapting or modifying the connection weights



in response to a pattern being presented at the input buffer and optionally the output buffer. A pattern presented at the output buffer corresponds to a desired response to a given input pattern. Therefore, this desired response must be provided by a knowledgeable "teacher". In such a case the learning task is referred to as supervised learning.

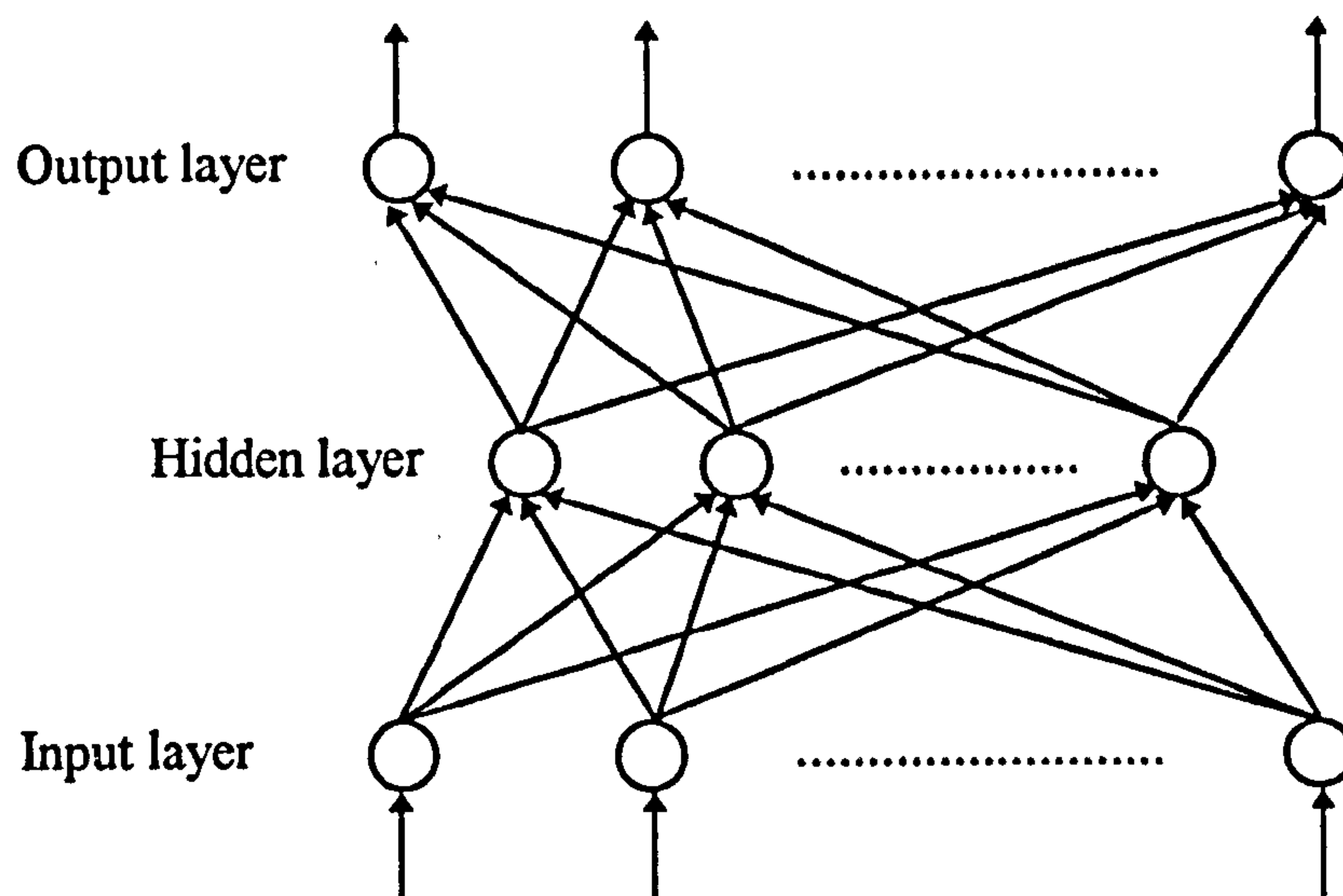


Figure 10.5 - A simple artificial neural network architecture

If the desired output is different from the input, the trained neural network is referred to as a hetero-associative network. If, for all training patterns, the desired output vector is equal to the input vector, the trained network is called auto-associative. If no desired output is shown the learning task is called unsupervised learning.

A third kind of learning falling between supervised and unsupervised learning is reinforcement learning where an external "teacher" indicates only whether the response to an input is good or bad. Whatever kind of learning is used, an essential characteristic of any artificial neural network is its learning rule which specifies how weights are adapted in response to a learning pattern. The learning procedure may require showing to an artificial neural network many examples, many thousands of times, until some convergence criterion is met or a pre-defined number of iterations is reached.

Recall has been used to refer to how the artificial neural network processes a pattern presented at its input buffer and creates a response at the output buffer. Often recall is an integral part of the learning procedure, such as when a desired response of the artificial neural network must be compared with the actual output of the network to generate an error signal. Such a situation occurs whenever a supervised learning strategy is followed.

The simplest form of a neural network has no feedback connections from one layer to another or to itself. Such a neural network is called a static "feedforward network" (Zbikowski and

Gawthrop 1992). In this case information is passed from the input buffer through intermediate layers to the output layer in a straightforward manner using the summation and transfer function characteristics of the particular neural network. Thus a feed forward artificial neural network performs a non-linear transformation of input data in order to approximate output data (Montague *et al.* 1992). Moreover, many types of artificial neural networks have an energy function associated with them, as presented with some detail in sub chapter 10.5. Each state of the network, which is defined by a particular set of processing element outputs, has an energy value. The recall procedure iteratively modifies the states so that energy decreases and, hence, a state representing a local minimum in the energy surface will be achieved.

The capabilities of the neural networks of the 1950s and 1960s were limited compared to those of modern neural network architectures. However, these early paradigms did bring many important properties to the attention of researchers. One of the most important paradigms is the so-called single layer perceptron, which is a neural network designed to learn to recognise simple patterns and is intended as a research tool for modelling possible brain mechanisms (Dayhoff 1990). The single layer perceptron pattern-mapping architecture learns to classify patterns through a supervised learning technique, and it is the simplest form of a artificial neural network used for classification of a special type of patterns said to be linearly separable, that is patterns that lie on the opposite sides of a hyperplane. Basically, the perceptron consists of a linear combiner followed by a hard limiter, as shown in Figure 10.6. Inputs arrive from the left hand side, and each incoming interconnection has an associated weight,  $w_{ji}$ . The perceptron processing element performs a weighted sum of its input values, and also accounts for an externally applied threshold,  $\theta$ . The

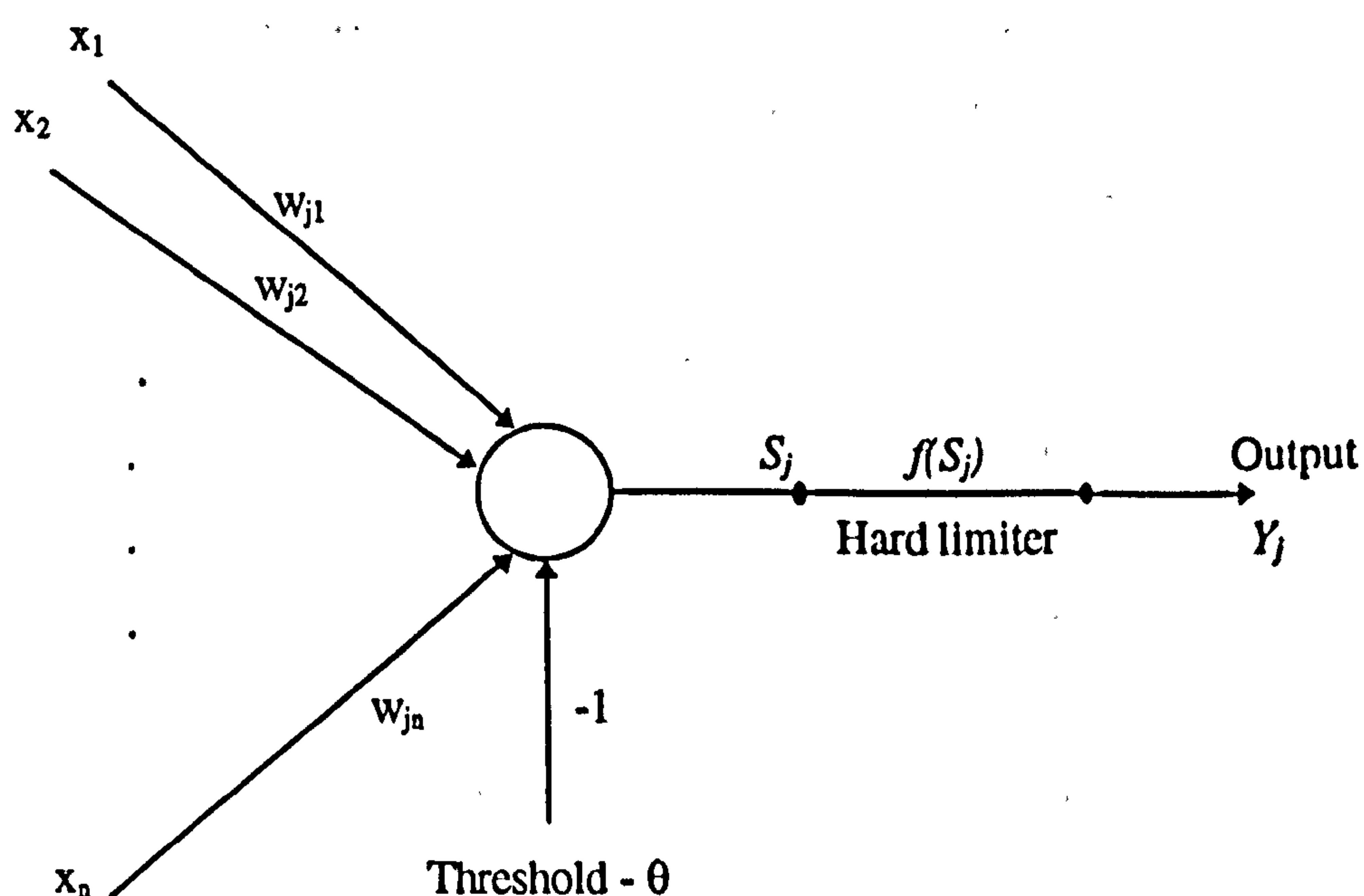


Figure 10.6 - Single layer perceptron



resulting sum is applied to a hard limiter such that the output is either +1, if the hard limiter input is positive, or -1, if the hard limiter input is negative.

From the model depicted in Figure 10.6, the linear combiner output, that is the hard limiter input, is given by the following equation,

$$S_j = \left( \sum_{i=0}^n a_i \times w_{ji} \right) - \theta \quad (10.1)$$

where the following notation is used,

$w_{ji}$  - weight associated with the connection to processing element  $j$ , which comes from processing element  $i$ ;

$a_i$  - value output of input element  $i$ ;

$S_j$  - weighted sum at processing element  $j$ ;

$\theta$  - threshold value.

A useful technique for analysing the behaviour of neural networks, such as the single layer perceptron, is to plot a map of the decision regions created in the multidimensional space spanned by the input variables. These decision regions specify which input values result in one class or another. The single layer perceptron forms two decision regions separated by a hyperplane (Lippmann 1987).

However, because the perceptron was a developmental device it had certain limitations. One, emphasised by Minsky and Papert (1969), was the inability to represent the basic Exclusive OR function. This is a result of the linear nature of the perceptron. A single layer perceptron can perform pattern classification only on linearly separable patterns. Linear separability requires that the patterns to be classified must be sufficiently separated from each other to ensure that the decision surfaces consist of hyperplanes (Haykin 1994). When inputs are not separable and distributions overlap, the decision boundaries may oscillate continuously with the original perceptron training procedure.

To overcome the single layer perceptron limitations, several advanced forms of artificial neural networks have been proposed. For example, the well known multilayer perceptron artificial neural network, which is in fact computationally more powerful than a single layer perceptron, can learn and categorise complex class categories. This is typically achieved by using processing elements with non-linear transfer functions, which are also referred to in the literature as activation functions. The fuzzy neural network used in the present fault diagnosis

approach, which is presented in the next sub chapter, is based on a multilayer perceptron architecture.

Multilayer perceptrons are feed forward neural networks with one or more layers of processing elements between the input and output nodes. These additional layers have been called hidden layers, since they are constituted by processing elements (hidden) which are not directly connected with the outside world. A general topology of a two layer perceptron neural network, which consists of one hidden layer, is depicted in Figure 10.7.

A multilayer perceptron is a tremendous step forward compared to its predecessor, the single layer perceptron, but was generally not used in the past because effective training algorithms were not available (Lippmann 1987). As quoted above, the single layer perceptron was limited to only two layers of processing elements, with only a single layer of adaptable weights. Therefore, this key limitation meant that the single layer perceptron could only classify patterns that were linearly separable. A multilayer perceptron overcomes this limitation because it can adjust two or more layers of connection weights, and uses a more sophisticated learning rule. The power of a multilayer perceptron neural network lies in its ability to train hidden layers and thereby escape the

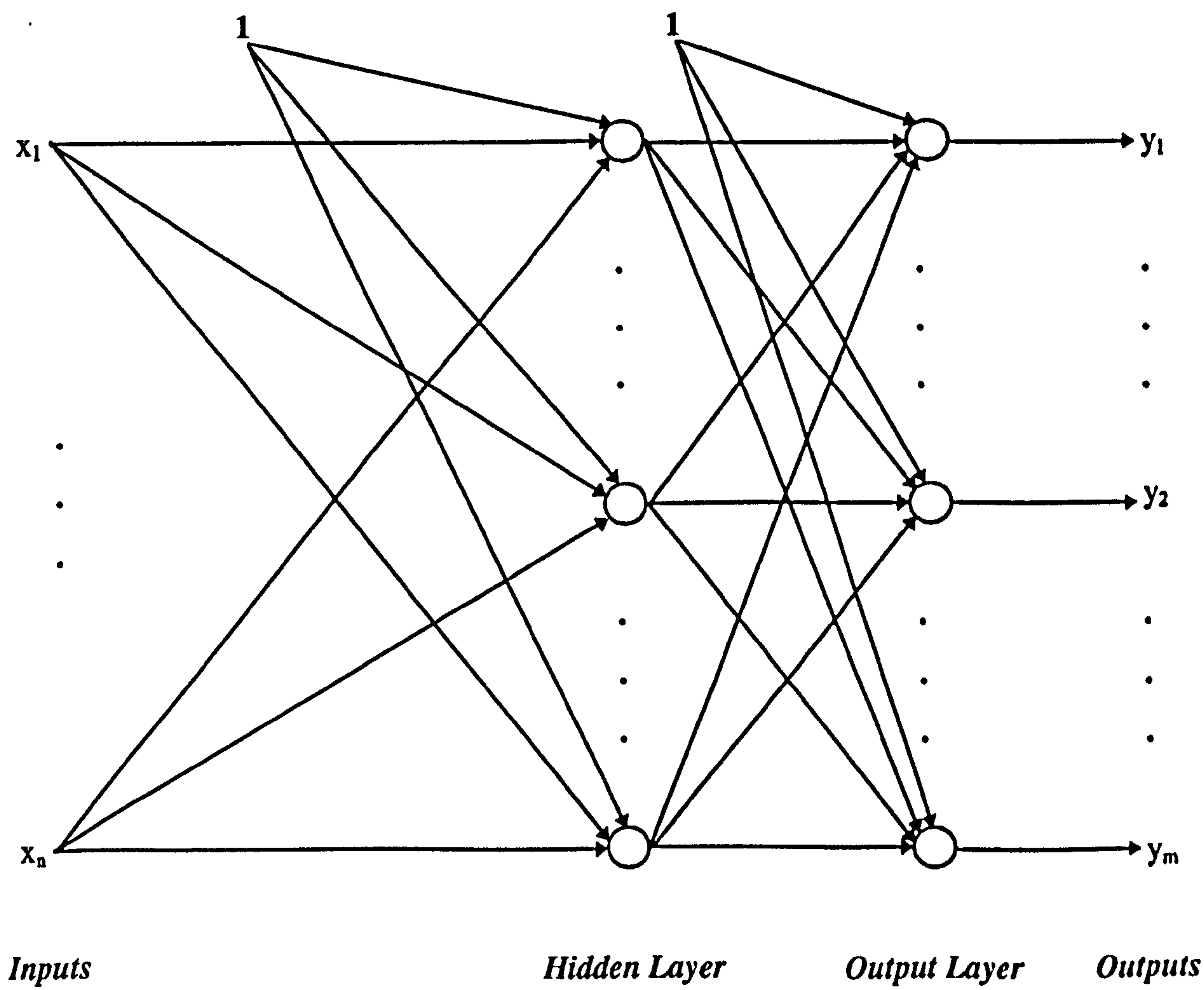


Figure 10.7 - General topology of a multilayer perceptron neural network



restricted capabilities of single layer neural networks. Moreover, the capabilities of multilayer perceptron neural networks stem from the non-linearities used as processing element activation functions. If these elements are linear, than a single layer neural network with appropriately chosen weights could exactly duplicate the calculations performed by any multilayer neural network (Lippmann 1987).

In multilayer perceptron neural networks each layer is fully connected to the succeeding layer. In the example shown in Figure 10.7, the arrows in the connections between processing elements are used to indicate the flow of information during recall. During the learning procedure, information is also propagated back through the neural network and used to update the connections weights. Moreover, such a neural network can be either hetero-associative or auto-associative. In the auto-associative approach the number of processing elements in the output layer is equal to the number of processing element in the input layer and, when the training task is performed, the input patterns are used also as the desired output values. This approach can be used for applications such as data compression or noise filtering. In the hetero-associative approach the output layer can have any number of processing elements, usually less than the input layer, and during the training procedure each example must be constituted by an input pattern, as well as the desired output values.

Artificial neural networks can be trained using a number of training methods. For instance, the backpropagation training algorithm (Rumelhart, Hinton and Williams 1986), the conjugate gradient algorithm (Leonard and Kramer 1990), or the genetic algorithm method (Goldberg 1989). However, for multilayer feed forward neural networks, the most popular training algorithm is the so called backpropagation learning algorithm or error backpropagation whose details are described in sub chapter 10.5.

## 10.4 Fuzzy Neural Networks

In the last few years the application of both technologies, neural networks and fuzzy logic, has received increasing attention in both research and application. Probably the first introduction of neural networks in consumer products occurred in Japan in December 1990, with a Matsushita air conditioner. Since then, a rapid growth in applications of neural networks and fuzzy logic in the consumer electronics industry has been observed in Japan. Moreover, a huge number of fuzzy logic industrial applications in Europe has also been reported (Altrock 1995).

A major reason for the widespread application of fuzzy systems in industry is that they have the ability to handle non-linear problems, are easy to understand, are easy to apply quickly,



and reduce development costs. However, fuzzy systems can express knowledge but cannot learn to adapt themselves. Neural networks have the ability to learn, so the two methods complement each other. There are five types of co-operative systems using both technologies: neural networks used as a development tool for building fuzzy systems; neural networks and fuzzy technology used independently; neural networks as correctors of the outputs of a fuzzy system; neural networks and fuzzy systems combined serially; and consumer-trainable functions (Takagi 1995).

In the present approach both techniques are combined serially for fault diagnosis purposes in industrial processes. Thus, for a neural network to be called a fuzzy neural network the signals and/or the weights must be fuzzy sets (Buckley and Hayashi 1994). From an engineering point of view much of the interest in neural networks and fuzzy systems has been for dealing with difficulties arising from uncertainty, imprecision and noise. Fuzzy reasoning is capable of handling uncertain and imprecise information while a neural network is capable of learning from examples. Fuzzy neural networks intend to combine the advantages of both fuzzy reasoning and neural networks. The fuzzy neural networks have been studied by many researchers and several different types of fuzzy neural networks have been proposed. Such types of fuzzy neural networks can be divided into three classes: fuzzy neural networks which have real number signals but fuzzy set weights; fuzzy neural networks which have fuzzy signals and real number weights; and the last class of fuzzy neural networks which have both fuzzy signals and fuzzy weights. In the present research work a fuzzy neural network of the second class has been used, that is the fuzzy neural networks, which have been used for fault diagnosis purposes, have fuzzy signals and real number weights.

This sub chapter describes a fuzzy neural network used for fault diagnosis purposes. Measurements or fault symptoms act as antecedents from which we can infer a classification of the pattern input, that is a diagnosis. In any classification task we have a measurement space from which we receive physical input data. In the present approach changes in the measurement process variables are received and then are linearly transformed into values in, say, the unit interval. Therefore, we have a normalised physical input space as an initial data representation. This means that after the linear transformation has been performed, the maximum change in a process variable, which can be observed, will be +1, and the minimum change will be -1. To perform this task it is necessary to define a parameter for each measurement variable which makes the normalisation of the different changes in the process variables in the range  $[-1, +1]$ . In previous chapters it has been reported that these normalisation parameters could affect the fault detection and diagnosis system performance, and should be set properly. However, since the training patterns used during the fuzzy neural network learning task have already been normalised, and there are no threshold values to trigger the system, the normalisation parameters do not affect the performance and reliability of the



fault detection and diagnosis approach proposed in this chapter.

Once the fuzzy neural network system has been triggered by the knowledge based fault detection block, which was described in the previous chapter, the normalised data is presented to the network. The fuzzy neural network topology, which is used in this approach, is depicted in Figure 10.8. According to the last sub chapter, it can be seen that the fuzzy neural network is achieved by adding a fuzzification layer to a conventional feed forward neural network. The fuzzification layer converts each input into the following quantity space,  $q_f = \{nlarge, nmedium, nsmall, zero, psmall, pmedium, plarge\}$ , by association with seven types of neurones in the fuzzification layer. The desired membership functions can be located by appropriately selecting the fuzzification layer weights of the fuzzy neural network.

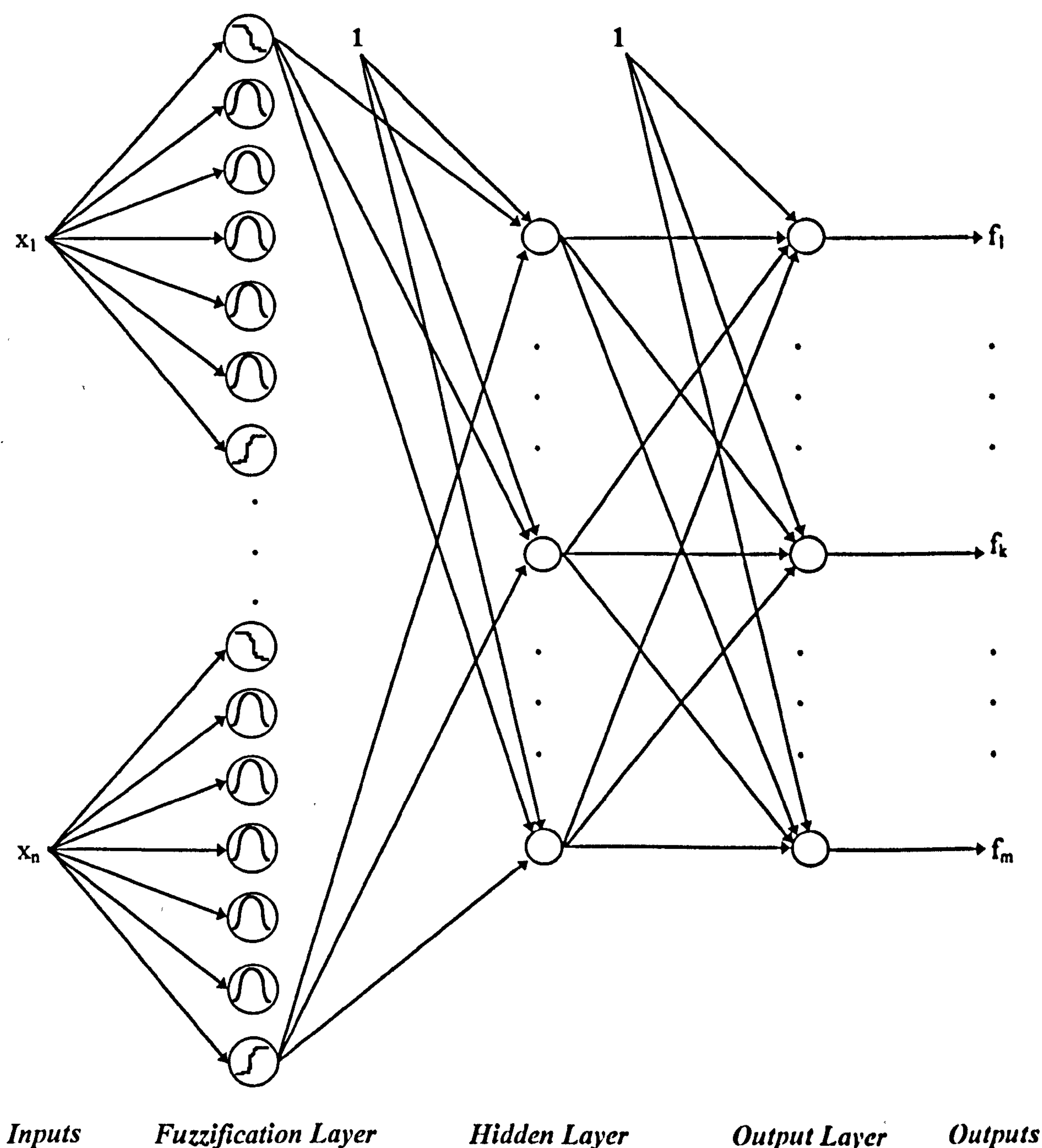


Figure 10.8 - Fuzzy neural network topology for fault diagnosis

A similar topology for a fuzzy neural network applied for fault diagnosis has already been reported by Zhang and Morris (1994), as well as a comparison between such a neural network and the classical feed forward neural network. In their paper some results are presented which demonstrate that a fault diagnosis system based on a fuzzy neural network approach performs much better than one based upon a conventional neural network. Therefore, comparing the architectures of both neural networks, we can conclude that the good performance of the fuzzy neural network is due to the additional fuzzification layer.

However, in the approach reported by Zhang and Morris, only three fuzzy sets have been used to discretize the fuzzy quantity space. As illustrated in Figure 10.8, in the present approach seven fuzzy sets have been used. The advantage of this approach is to allow a reduced set of measurement variables to form the input space of the fuzzy neural network and increase the number of faults that can be diagnosed. Hence, in the approach presented in this report, the amount of information handled in the diagnosis task is reduced and the number of possible patterns to recognise can be increased.

The processing elements of the fuzzification layer associated with the fuzzy sets *nlarge* and *plarge* use the complement sigmoid function and the sigmoid function, respectively, as their activation functions. Such an activation functions, which define the outputs of a processing elements in terms of the activity level at their inputs, can be represented by the following two equations, respectively. The corresponding shapes of these activation functions are shown in Figures 10.9 and 10.10, respectively.

$$f(z) = 1 - \frac{1}{1 + e^{-w_i z}} \quad (10.2)$$

$$f(z) = \frac{1}{1 + e^{-w_i z}} \quad (10.3)$$

where the following notation is used,

- $f(z)$ , is the processing element output;
- $z$ , is the processing element input;
- $w_i$ , is the input weight.

The other processing elements of the fuzzification layer associated with the remaining fuzzy sets use the gaussian function as their activation function. By using the same notation mentioned above, for the complement sigmoid and sigmoid functions, the gaussian activation function can be



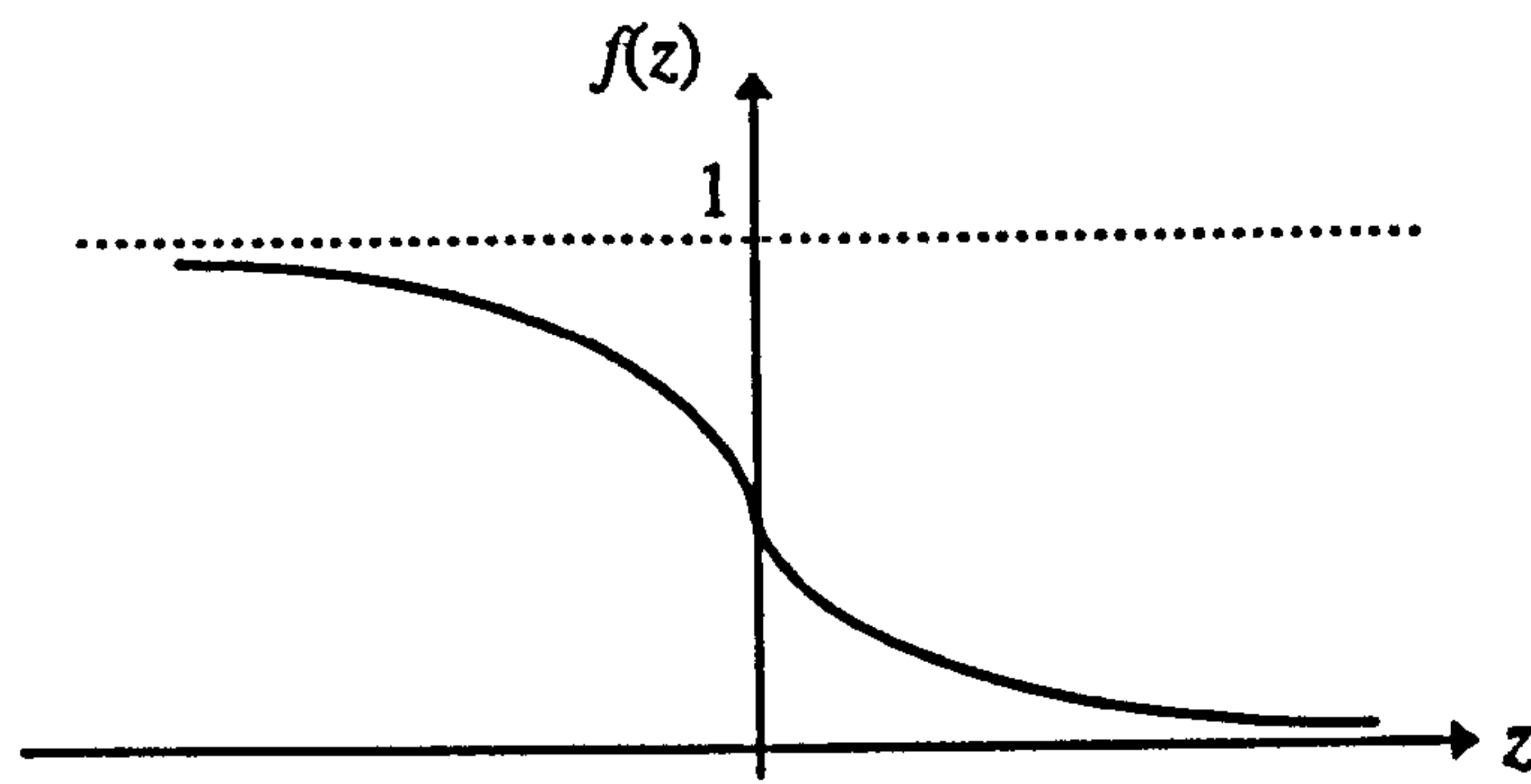


Figure 10.9 - Complement sigmoid function

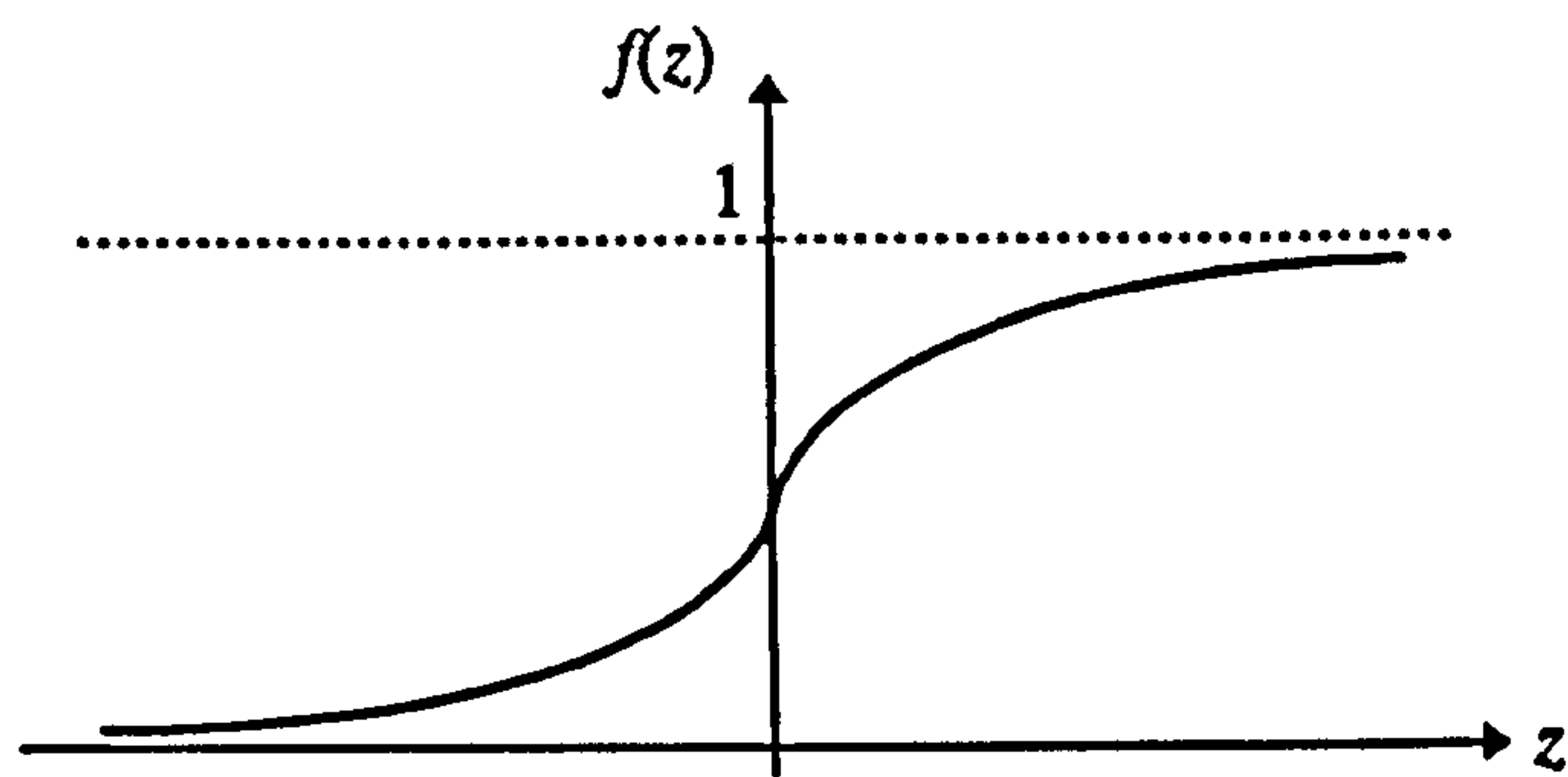


Figure 10.10 - Sigmoid function

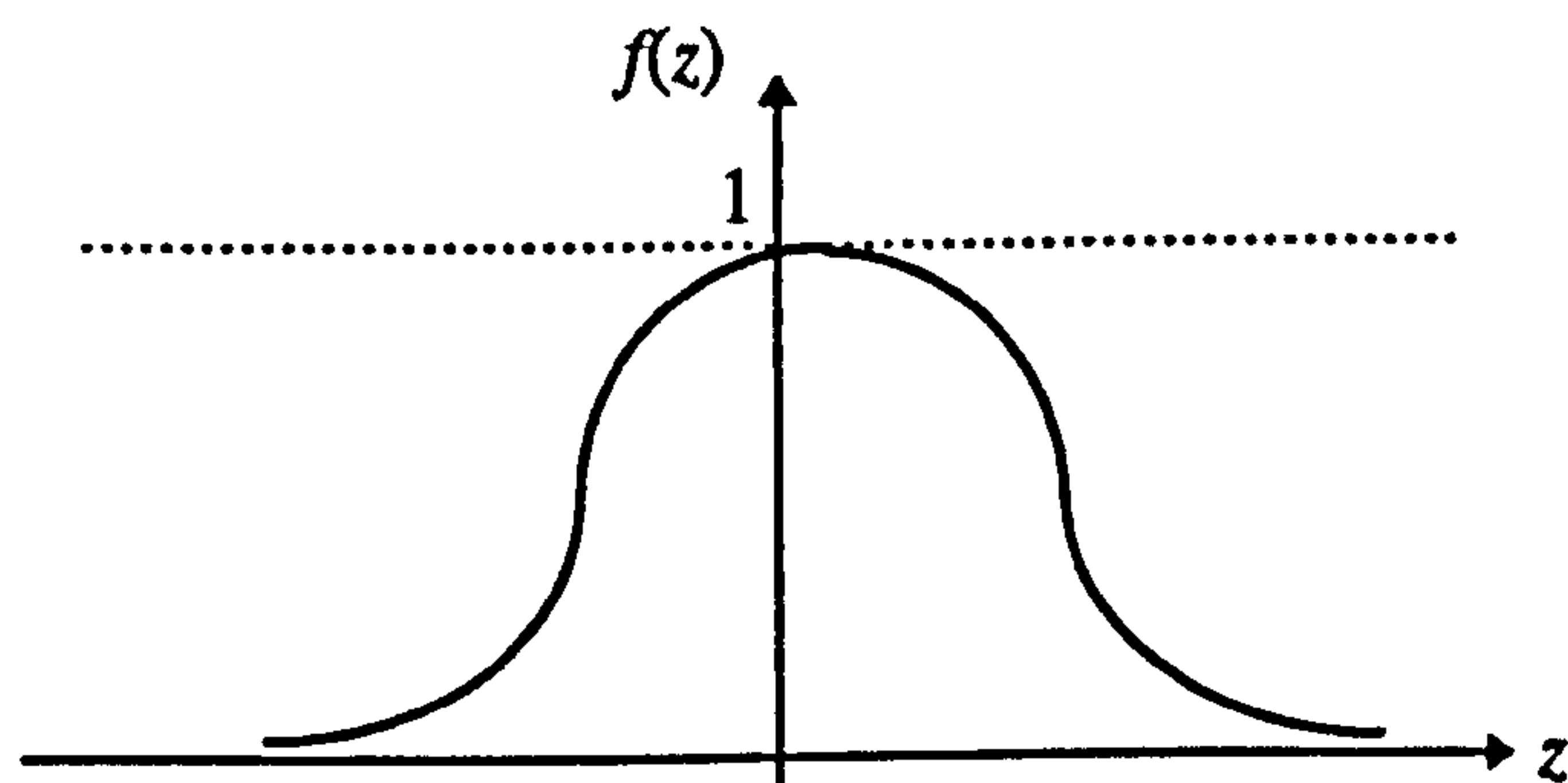


Figure 10.11 - Gaussian function

represented by the following equation, (10.4), and the corresponding shape of this activation function is given in Figure 10.11.

$$f(z) = e^{-(w_i z)^2} \quad (10.4)$$

The processing elements in the hidden and output layers use the sigmoid function. For these processing elements the variable,  $z$ , quoted above, represents the weighted sum of all the inputs of each processing element. As shown in Figure 10.8, each output of the fuzzy neural network is used to represent a particular fault. Outputs of the fuzzy neural network take values in the range zero to

one and a fault is indicated when a corresponding network output is close to 1. Therefore, if for a specific input pattern there is no network output taking a value close to 1, then that input pattern does not represent a faulty scenario. However, if more than one output of the fuzzy neural network takes a value close to 1, this means that more than one fault has occurred in the process under concern.

Training data are on-line measurements covering the events of the faults being considered and the nominal operating conditions. These can be obtained from a recorded operating history of the process or, as in the applications presented in sub chapters 10.6 and 10.7, from simulation studies. The weights of the fuzzification layer have been initialised based on experiences about the process under consideration, such as nominal fluctuation levels of process variables. The fuzzy neural network proposed here can be seen as a feed forward neural network, since there are no feedback connections in the layers or between them. As quoted above, for such neural networks the most popular training algorithm is the so-called backpropagation method. However, during this research work an extension of such a learning algorithm has been developed, which is presented in the next sub chapter. This extended backpropagation learning algorithm has the advantage of decreasing the learning time to achieve a pre-defined accuracy in the network outputs, as can be seen through the results presented in sub chapter 10.5.

The fault diagnosis system based on this fuzzy neural network approach has been applied successfully in simulation studies of the mixing process and a continuous stirred tank reactor. In these processes single and double simultaneous abrupt faults have been considered, as well as incipient faults. During these studies, it has been observed that the neural network's generalisation ability has the major importance in the diagnosis of incipient faults, since the training patterns used only include abrupt faults symptoms. A detailed description of the results achieved is given in sub chapters 10.6 and 10.7, respectively, for the mixing process and for the continuous stirred tank reactor.

## **10.5 Backpropagation Training**

Among the many interesting properties of an artificial neural network, the property that is of primary significance is the ability of the network to learn from its environment and to improve its performance through learning; the improvement in performance taking place over time in accordance with some prescribed measure. In the present study the neural network learns about its environment through an iterative process of adjustments applied to its synaptic weights. The learning procedures are usually divided into three classes: supervised learning, reinforcement



learning and self-organised or unsupervised learning. Supervised learning is performed under the supervision of an external "teacher"; reinforcement learning involves the use of a "critic" that evolves through a trial and error process; unsupervised learning is performed in a self-organised manner in that no external teacher or critic is required to instruct synaptic changes in the network (Haykin 1994).

As discussed in previous sub chapters, a feed forward neural network is used in the study presented in this chapter, since there are no interconnections between the output of a processing element and the input of a processing element in the same layer or in a preceding layer. The so called backpropagation, or in other words the error back propagation algorithm, is the most widely used learning technique applicable to feed forward neural networks (Turban 1992). An extension of such a learning technique is used in the present approach. Hence, as under this learning method the fuzzy neural network weights are adjusted each time a correct pattern, represented by an input-output example, is externally provided, we can say that a supervised learning procedure is used in this research work. Each example requires two stages: a forward pass and a backward pass. The forward pass involves presenting a sample input, from a set of input-output examples, to the neural network and letting activation flow until the output layer is reached. During the backward pass, the actual output of the neural network is compared with the desired or target output for that sample input. The errors between the actual and desired outputs are used to adjust the weights for the connections to the previous layer. We can then use the output unit errors to derive the hidden layer output errors. Finally, errors are propagated back to the connections coming from the input units.

The error back propagation algorithm is based on an error correction learning rule and it may be viewed as a non-linear extension of an equally popular adaptive filtering algorithm, the so called least-mean-square (LMS) method which is also known as the delta rule or the Widrow-Hoff rule, developed for a single linear neurone model (Haykin 1994). However, one of the problems associated with this algorithm is the training time needed to achieve a pre-defined accuracy in the artificial neural network outputs. From the fault diagnosis point of view, this problem is specially important when we have a large number of faulty scenarios for the process under concern. Therefore, in order to speed up the backpropagation learning technique an extension of the standard error back propagation algorithm is developed, which is presented in the remainder of this sub chapter. Such an algorithm is applied to the mixing process, as well as to the continuous stirred tank reactor and, a results comparison is made between the standard backpropagation algorithm and the extended one proposed below.

Suppose that the neural network has some global error function  $E$ , with an associated differentiable function of all the connections weights in the neural network. Hence, the critical parameter that is passed back through the layers is defined by equation (10.5),

$$e_j^{[s]} = -\frac{\partial E}{\partial I_j^{[s]}} \quad (10.5)$$

where the following notation is used,

$e_j^{[s]}$  - current error of  $j$ th neurone in layer,  $s$ ;

$E$  - global error function;

$I_j^{[s]}$  - weighted summation of inputs to  $j$ th neurone in layer,  $s$ .

The result of equation (10.5) can be considered as a measure of the local error at processing element  $j$ , in the level  $s$ . Using the chain rule twice in succession gives a relationship between the local error at a particular processing element at level  $s$ , and all the local errors at the level  $s+1$  (Haykin 1994), as shown in equation (10.6),

$$e_j^{[s]} = f'(I_j^{[s]}) \times \sum_k (e_k^{[s+1]} \times w_{kj}^{[s+1]}) \quad (10.6)$$

where  $e_j^{[s]}$  and  $I_j^{[s]}$  are defined above and the other variables are defined as follows,

$e_k^{[s+1]}$  - current error of  $k$ th neurone in layer  $s+1$ ;

$w_{kj}^{[s+1]}$  - weight on connection joining  $j$ th neurone in layer  $s$  to  $k$ th neurone in layer  $s+1$ .

Note that in equation (10.6) there is a layer above layer  $s$ , therefore this equation can only be used for non-output layers.

If the activation function,  $f$ , considered for a processing element, is the sigmoid function as defined by equation (10.3), then its derivative can be expressed as a simple function of itself as follows,

$$f'(z) = f(z) \times (10 - f(z)) \quad (10.7)$$

where the use of prime (on the left hand side) signifies differentiation with respect to the argument. Moreover, a processing element (neurone) transfers its inputs as expressed by the following equation,



$$x_j^{[s]} = f\left(\sum_i (w_{ji}^{[s]} \times x_i^{[s-1]})\right) = f(I_j^{[s]}) \quad (10.8)$$

where the following notation is used,

$x_j^{[s]}$  - current output state of  $j$ th neurone in layer  $s$ .

Therefore, from equations (10.7) and (10.8), equation (10.6) can be rewritten as follows,

$$e_j^{[s]} = x_j^{[s]} \times (1.0 - x_j^{[s]}) \times \sum_k (e_k^{[s+1]} \times w_{kj}^{[s+1]}) \quad (10.9)$$

Thus the main mechanism in a back-propagation network is to forward propagate the input through the layers to the output layer, determine the error at the output layer, and then propagate the errors back through the network from the output layer to the input layer. The multiplication of the error by the derivative of the transfer function scales the error. The transfer function derivative serves to keep the error correction well bounded. The weights of each input to the  $j$ th neurone are then adjusted in proportion to this calculated error. The aim of the learning process is to minimise the global error  $E$  of the system by modifying the weights.

Given the current set of weights  $w_{ji}^{[s]}$ , we need to determine how to increment or decrement them in order to decrease the global error. This can be done using a gradient descent rule which can be expressed by the following equation,

$$\Delta w_{ji}^{[s]}(k+1) = -lcoef \times \frac{\partial E}{\partial w_{ji}^{[s]}} \quad (10.10)$$

where  $lcoef$  is a learning coefficient; and  $k$  is the adaptation step. In other words, each weight is changed according to the size and direction of negative gradient on the error surface. The partial derivative in equation (10.10) can be calculated directly from the local error values, because, by the chain rule and equation (10.8), we have,

$$\frac{\partial E}{\partial w_{ji}^{[s]}} = \frac{\partial E}{\partial I_j^{[s]}} \times \frac{\partial I_j^{[s]}}{\partial w_{ji}^{[s]}} = -e_j^{[s]} \times x_i^{[s-1]} \quad (10.11)$$

Substituting equation (10.11) into equation (10.10), gives,

$$\Delta w_{ji}^{[s]}(k+1) = lcoef \times e_j^{[s]} \times x_i^{[s-1]} \quad (10.12)$$

The above discussion has assumed the existence of a global error function without actually specifying it. This function is needed to define the local errors at the output layer so that they can be propagated back through the network. Suppose that a vector  $\underline{i}$  is presented at the input edge layer of the network and suppose that the desired output  $\underline{d}$  is specified by a teacher. Let  $\underline{o}$  denote the actual output produced by the network with its current set of weights. Then an iterative gradient algorithm is used to minimise a cost function equal to the mean square error between the desired and the actual neural network outputs (Lippmann 1987), which can be defined as,

$$E = \frac{1}{2} \times \sum_k (d_k - o_k)^2 \quad (10.13)$$

where the subscript  $k$  indexes the components of  $\underline{d}$  and  $\underline{o}$ . From equation (10.5) the scaled "local error", at each processing element of the output layer, is given by,

$$e_k^{(o)} = -\frac{\partial E}{\partial I_k^{(o)}} = -\frac{\partial E}{\partial o_k} \times \frac{\partial o_k}{\partial I_k} = (d_k - o_k) \times f'(I_k) \quad (10.14)$$

One of the problems of the algorithm discussed above is setting an appropriate learning rate. Changing the weights as a linear function of the partial derivative as defined in equation (10.10) makes the assumption that the error surface is locally linear, where "locally" is defined by the size of the learning coefficient. However, at points of high curvature this linearity assumption does not hold and divergent behaviour might occur near such points. Therefore, it is important to keep the learning coefficient low to avoid such behaviour.

On the other hand, a small learning rate coefficient can lead to very slow learning. The concept of "momentum term" was introduced to resolve this dichotomy. Hence, the delta weight equation (10.12) is modified so that a portion of the previous delta weight is fed through to the current delta weight (Rumelhart 1986a), as it is shown by the following equation,

$$\Delta w_{ji}^{[s]}(k+1) = lcoef \times e_j^{[s]} \times x_i^{[s-1]} + momentum \times \Delta w_{ji}^{[s]}(k) \quad (10.15)$$

This acts as a low-pass filter on the delta weight terms since general trends are reinforced where as oscillatory behaviour cancels itself out. This allows a low learning coefficient but faster learning.



In order to speed up the back propagation learning technique a number of methods have been reported (Fahlman 1988, Leonard and Kramer 1990, Haykin 1994). These include changing the learning rate from one iteration to the next. For instance, Quickprop (Fahlman 1988) is a method which uses the curvature of the error surface to speed up the learning task. Quickprop assumes the error surface to be locally quadratic and attempts to jump in one step from the current position directly in the minimum of the parabola. Quickprop computes the derivatives in the direction of each parameter and after computing the first gradient, with regular backpropagation a direct step to the error minimum is attempted.

However, one simple technique, which almost doubles the speed, is to add a small positive offset,  $f'_{offset}$ , to the derivative of the sigmoid function, prior scaling the local error. Following this procedure, equations (10.6) and (10.14) can be rewritten, respectively, as follow,

$$e_j^{[s]} = \left[ f'(I_j^{[s]}) + f'_{offset} \right] \times \sum_k (e_k^{[s+1]} \times w_{kj}^{[s+1]}) \quad (10.16)$$

$$e_k^{(o)} = -\frac{\partial E}{\partial I_k^{(o)}} = -\frac{\partial E}{\partial o_k} \times \frac{\partial o_k}{\partial I_k} = (d_k - o_k) \times [f'(I_k) + f'_{offset}] \quad (10.17)$$

The rationale for this is that when the incoming weights of a processing element become large, the summation values are large and, hence, the activation values become saturated (0.0 or 1.0). When such a situation happens, the derivative becomes zero and the scaled local error is always zero. Thus the processing element stops learning. Adding a positive offset to the derivative alleviates this problem. In the remainder of this sub chapter some results achieved during the current studies are presented, which demonstrate the power of the derivative offset parameter in increasing the speed of the neural network training procedure.

The standard backpropagation learning algorithm, as well as the extended one proposed above have been applied to train a fuzzy neural network for fault diagnosis purposes in the mixing process and in the continuous stirred tank reactor respectively. Training data was obtained from previous simulation studies of the process under concern, with the aim of covering all the faulty scenarios being considered and the nominal operational conditions. Moreover, as a fault can have more than one fault symptom according to the different possibilities for the steady state condition of the process, the number of patterns that the network should recognise is always higher than the

number of faults being considered. Therefore, for the mixing process, 382 patterns have been used as training data of the fuzzy neural network, while for the continuous stirred tank reactor 105 patterns have been used. These patterns are divided essentially into three groups, one corresponding to single abrupt fault symptoms, another covering double simultaneous abrupt fault scenarios and the last one covering the nominal operational conditions. The set of training patterns do not include incipient fault symptoms. However, as neural networks have the ability to generalise, it is expected that such a fuzzy neural network can diagnose incipient faults. Some results achieved during simulation studies, which have been conducted with the processes quoted above, are presented in subsequent sub chapters.

The training procedure followed during the present studies can be divided into two stages. In the first stage the training patterns have been presented to the fuzzy neural network a pre-defined number of times. Data is read randomly and presented to the neural network as an input/output pair. Therefore, through the learning algorithm presented above, the connection weights between the fuzzification layer and the hidden layer, as well as the connection weights between the hidden layer and the output layer, are adjusted in order to minimise the error between the actual network outputs and the desired ones. Thus, when the pre-defined number of iterations is reached, this phase is terminated. Moreover, before this phase begins, the learning coefficient, the momentum term and the derivative offset parameter should be defined. In the second stage all the training patterns used during the learning task are sequentially presented to the network. No weights adaptation is made during this phase, but the errors between the actual network outputs and the desired ones are retained.

The fuzzy neural network, which has been used for fault diagnosis purposes in the mixing process, has 42 processing elements in the fuzzification layer corresponding to 6 measurement variables, 20 hidden processing elements in the hidden layer and 6 processing elements in the output layer corresponding to 6 faults being considered. Thus, with the following parameter values pre-defined:

- Number of iterations, 20000;
- Learning coefficient, 0.9;
- Momentum term, 0.6;
- Derivative offset, 0;



the results achieved, after the two stage training procedure has been performed, are depicted in Figure 10.12. Note, since the derivative offset parameter was selected as 0, according to equations (10.16) and (10.17), this trial corresponds to the situation where the standard backpropagation learning algorithm has been used.

A maximum error signal of 0.249 is shown in Figure 10.12. It appears in the output 2 of the fuzzy neural network when pattern 364 is presented. In order to improve the performance of the learning procedure, the derivative offset parameter has been introduced. Thus, another trial has

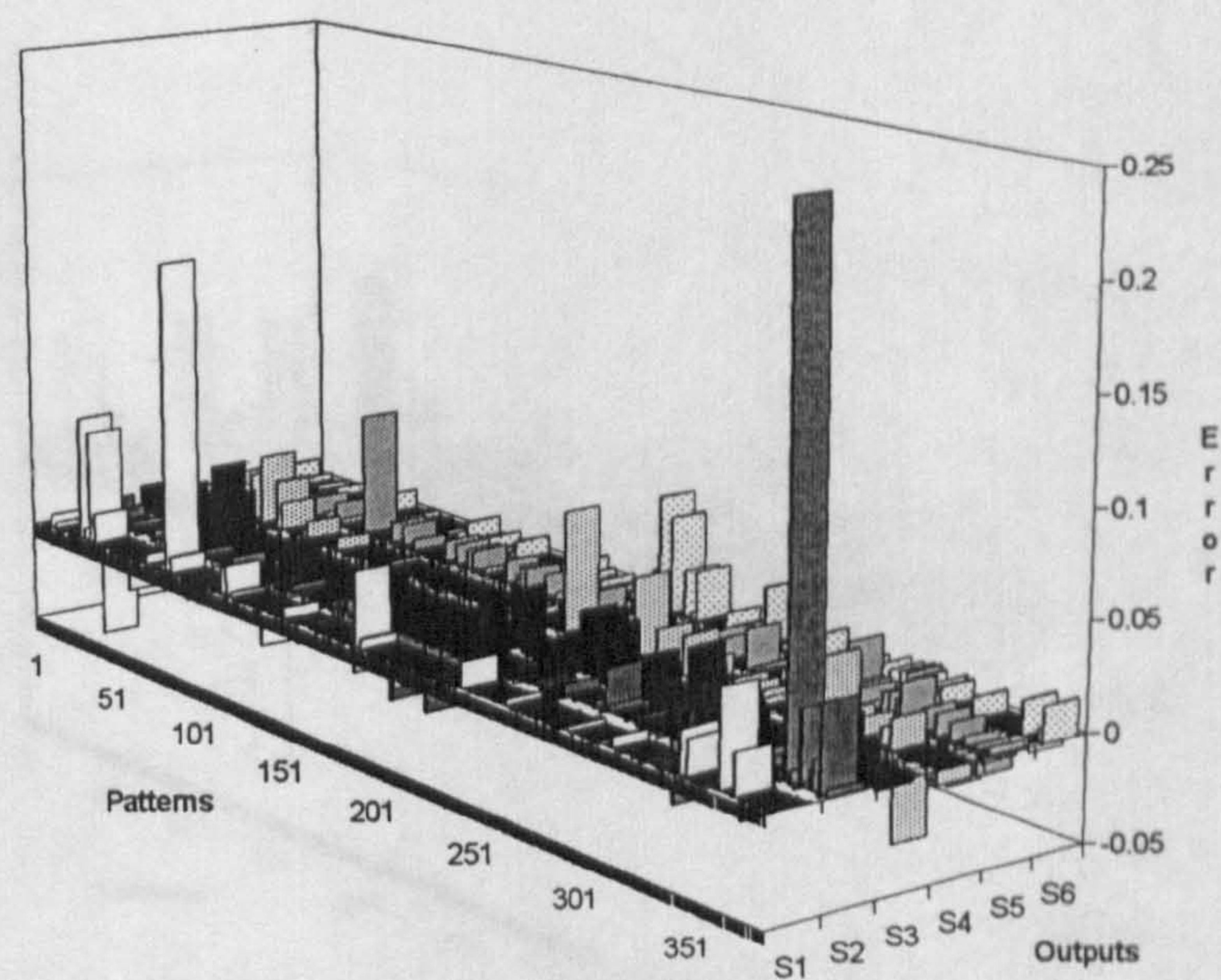


Figure 10.12 - FNN output errors for the mixing process (Derivative offset = 0)

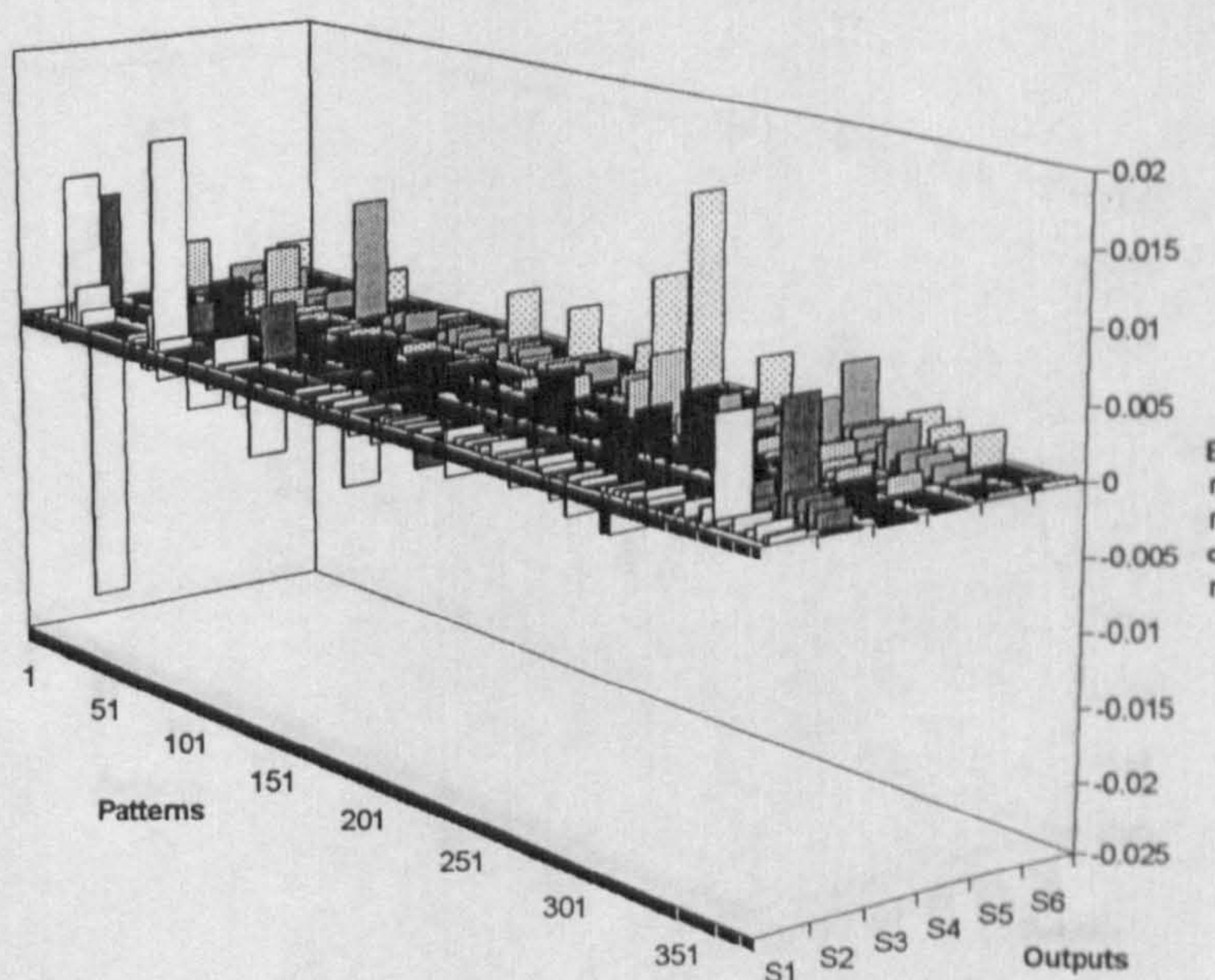


Figure 10.13 - FNN output errors for the mixing process (Derivative offset = 0.19)



been made with the derivative offset equal to 0.19 and the other parameters having the values quoted above. The results obtained after the fuzzy neural network training task was completed are shown in Figure 10.13.

Figure 10.13 shows the improvements achieved with the introduction of the derivative offset parameter. The maximum error signal has been reduced almost ten times. In order to determine the derivative offset value which gives the best result, further trials have been made. For instance, Figures 10.14 and 10.15 depict the results obtained with the derivative offset parameter equal to 0.2 and 0.21, respectively, and the other parameters still having the same values as in previous trials.

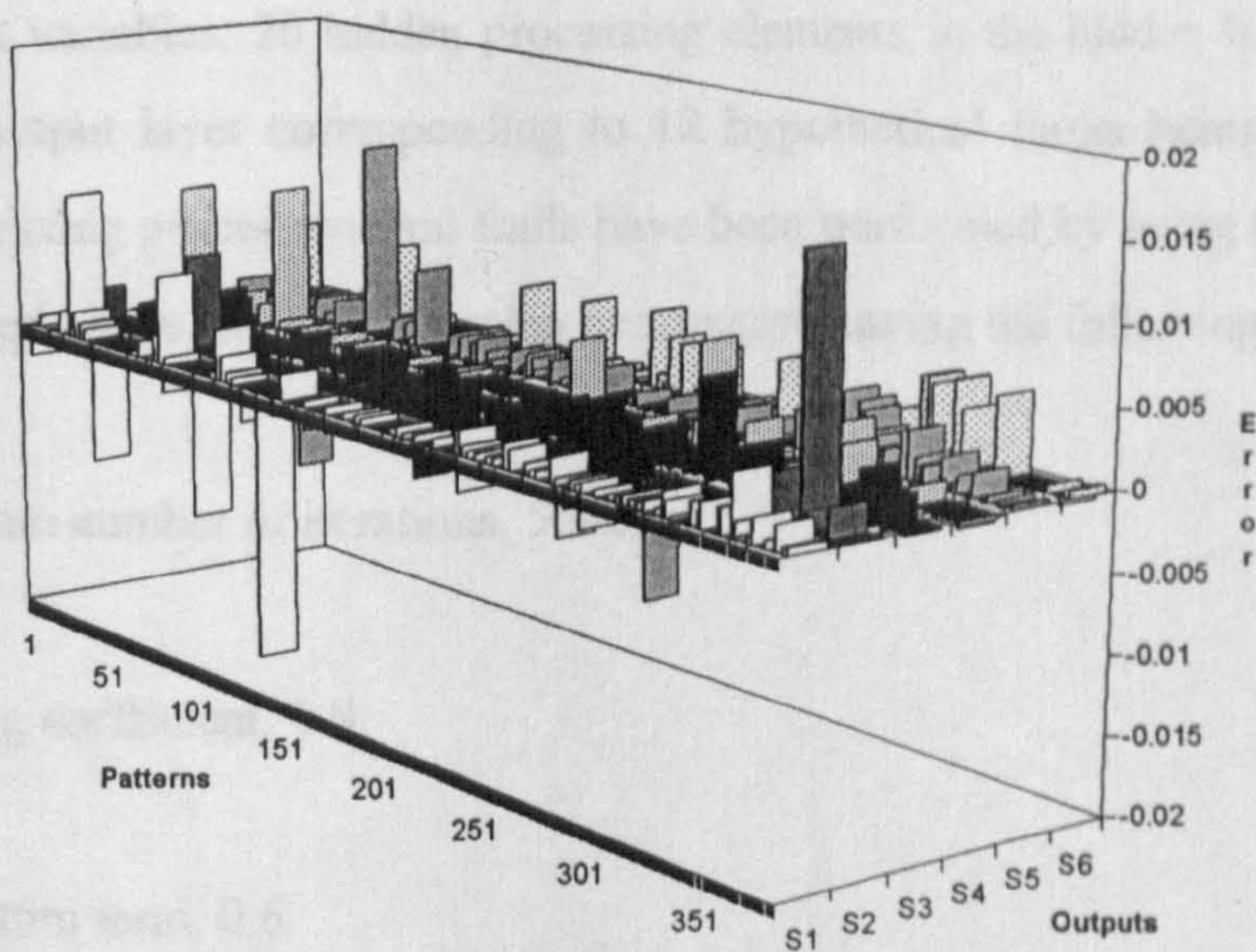


Figure 10.14 - FNN output errors for the mixing process (Derivative offset = 0.2)

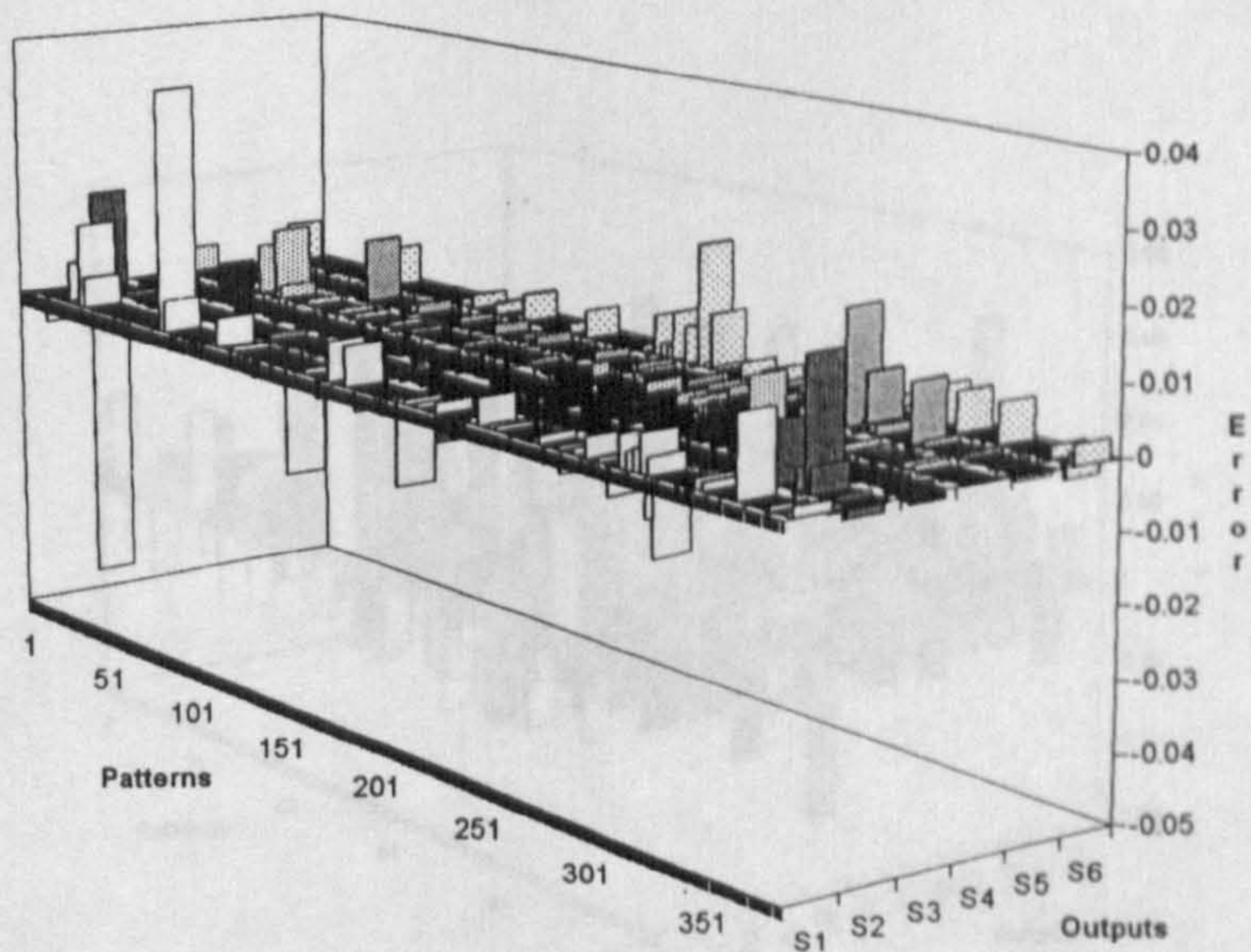


Figure 10.15 - FNN output errors for the mixing process (Derivative offset = 0.21)



Comparing the results depicted in Figures 10.12 to 10.15, one can see the importance of the derivative offset parameter in the training task of a neural network. The best results have been obtained with the derivative offset equal to 0.2, as shown in Figure 10.14. Hence, the fuzzy neural network, which was achieved after the learning procedure has been terminated, has been implemented for fault diagnosis purposes. The results achieved with such a fuzzy neural network, are presented in the following sub chapters. Moreover, under the above considerations the learning task has been taken an average time of forty minutes, in a 386 PC (25 MHz) fitted with a mathematical co-processor.

For the continuous stirred tank reactor a fuzzy neural network, with the same topology described above and consisting of 63 processing elements in the fuzzification layer corresponding to 9 measurement variables, 20 hidden processing elements in the hidden layer and 12 processing elements in the output layer corresponding to 12 hypothetical faults being considered, has been used. As for the mixing process several trails have been performed by using the supervised learning algorithm presented above, with the learning parameters having the following values,

- Maximum number of iterations, 5000;
- Learning coefficient, 0.9;
- Momentum term, 0.6.

Figures 10.16 and 10.17 give the worst and the best results obtained, respectively with derivative offset equal to 0 and equal to 0.13. Once again the results achieved when the derivative

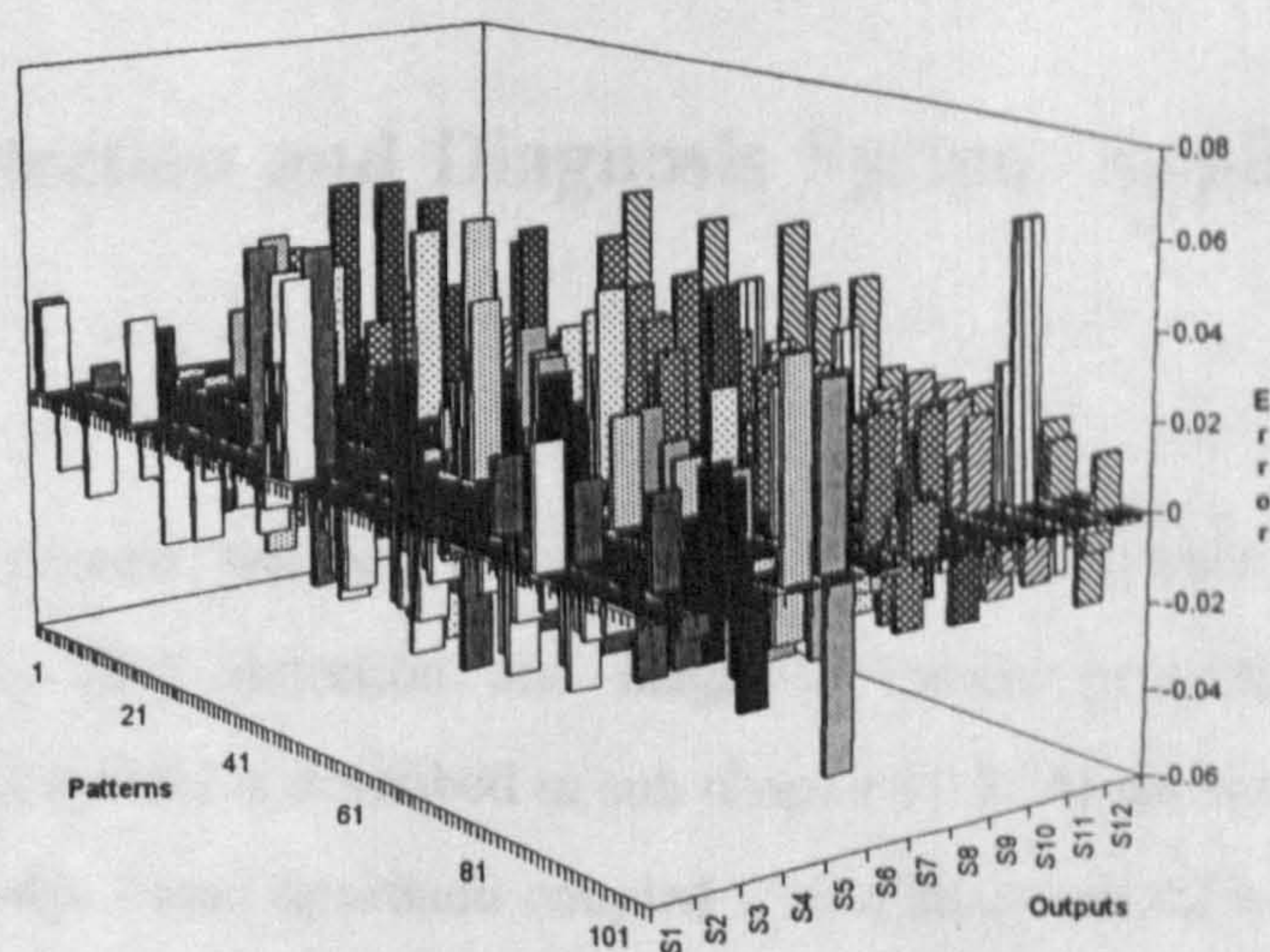


Figure 10.16 - FNN output errors for the CSTR (Derivative offset = 0)



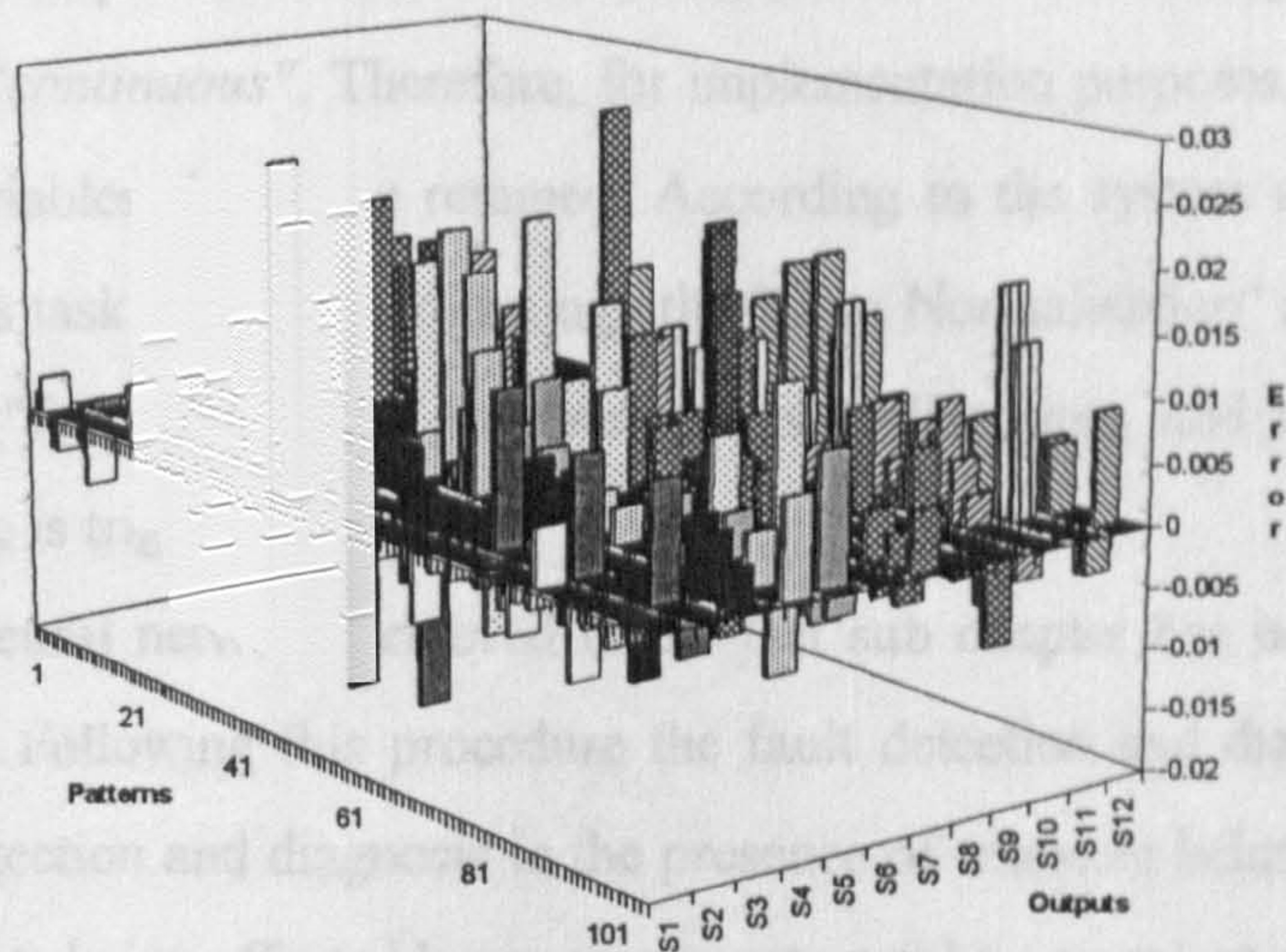


Figure 10.17 - FNN output errors for the CSTR (Derivative offset = 0.13)

offset parameter is used are much better than the results achieved with the derivative offset having the value zero. The error has been reduced almost three times. This suggests that if we are concerned with neural network training time the derivative offset parameter should be used. For the continuous stirred tank reactor, the fuzzy neural network learning task has been taken an average time of seventeen minutes, in the same computer quoted above. After the training procedure has been completed, a fault diagnosis system based on such a fuzzy neural network has been implemented.

The fuzzy neural networks just achieved for the mixing process and for the continuous stirred tank reactor, respectively, have been applied for fault diagnosis purposes in simulation studies conducted with each process. In the next two sub chapters the results achieved during such simulation studies are presented.

## 10.6 Fault Detection and Diagnosis System Applied to the Mixing Process

The mixing process, which is described in chapter 4., has been used as a test bed of the distributed intelligent fault detection and diagnosis system presented in this chapter. The architecture of such a system is described in sub chapter 10.2. As quoted above the overall system consists of a knowledge based approach coupled with a fuzzy neural network. Fault detection is performed through the knowledge based system where fault detection heuristic rules have been generated from deep and shallow knowledge of the mixing process as described in the previous



chapter. From this description, it can be seen that in some heuristic rules pre-condition, is used the linguistic statement "*continuous*". Therefore, for implementation purposes, the last two changes in the measurement variables should be retained. According to the system architecture described in sub chapter 10.2, this task is performed through the "Data Normalisation" module which passes the retained data to the "Fault Detection" module each sampling time, and to the "Fault Diagnosis" module when this one is triggered.

The fuzzy neural network achieved in the last sub chapter has been used to perform the fault diagnosis task. Following this procedure the fault detection and diagnosis system can cope with on-line fault detection and diagnosis in the presence of transient behaviours without the fuzzy neural network outputs being affected by measurement variables transient response.

Single and double simultaneous abrupt faults, which are represented in Tables 6.5 and 7.1 respectively, have been considered. Moreover, as quoted in the last sub chapter, six measurement variables are used as input data to the fuzzy neural network which performs the diagnosis task. These measurement variables are the following:

- $L_1$ , level in tank 1;
- $T_1$ , temperature in tank 1;
- $L_2$ , level in tank 2;
- $T_2$ , temperature in tank 2;
- $Q_c$ , input cold water flow rate;
- $Q_h$ , input hot water flow rate.

However, as the linguistic statement "*continuous*" is used in the knowledge based system responsible for triggering the fault diagnosis module, at least a hypothetical fault could only be diagnosed in the second sampling time after the fault has occurred. Thus, since  $Q_c$  and  $Q_h$  used as inputs of the fuzzy neural network are also used to simulate some faults, in the second sampling time after such abrupt faults have occurred, no change will be observed in these two measurement variables. Hence, following this procedure, the initial set of six measurement variables used to achieve the fault symptoms will be reduced to four variables. However, during previous research work carried out with the mixing process, which is presented in chapter 7., it was found that only four variables are not enough to diagnose all the faults quoted above. Therefore, in order to overcome this problem, the fuzzy neural network input data is constituted by the changes in the measurement variables  $L_1$ ,  $L_2$ ,  $T_1$  and  $T_2$  observed when the fault diagnosis module is triggered, together with the changes in the measurement variables  $Q_c$  and  $Q_h$  observed at a previous sampling.



According to the previous sub chapters, the use of six measurement variables as input data of the fuzzy neural network implies the existence of 42 processing elements in the fuzzification layer, arranged in six groups corresponding to the six on-line information sources. The number of processing elements in the hidden layer is determined by the complexities of the relationships between the faults and the fault symptoms. Therefore, during the current studies, it is found that 20 hidden processing elements could give good performance and, thus, the number of processing elements in the hidden layer was fixed at 20. From Table 6.5, it can be seen that six single faults have been considered. Moreover, as the double simultaneous faults have been achieved through an AND operation in the single fault space, the number of processing elements in the output layer was fixed at 6 each one corresponding to a fault. As quoted in the last sub chapter, training data are obtained from previous simulation studies. All single and double simultaneous abrupt faults have been initiated with all the possible combinations of the setpoint values, and then 382 patterns have been achieved corresponding to the fault symptoms and the nominal operating conditions. As described in sub chapter 10.5, several trials have been performed to achieve the fuzzy neural network used in the present approach.

During simulation studies conducted with the present fault detection and diagnosis approach applied to the mixing process, all the possible single and double simultaneous abrupt

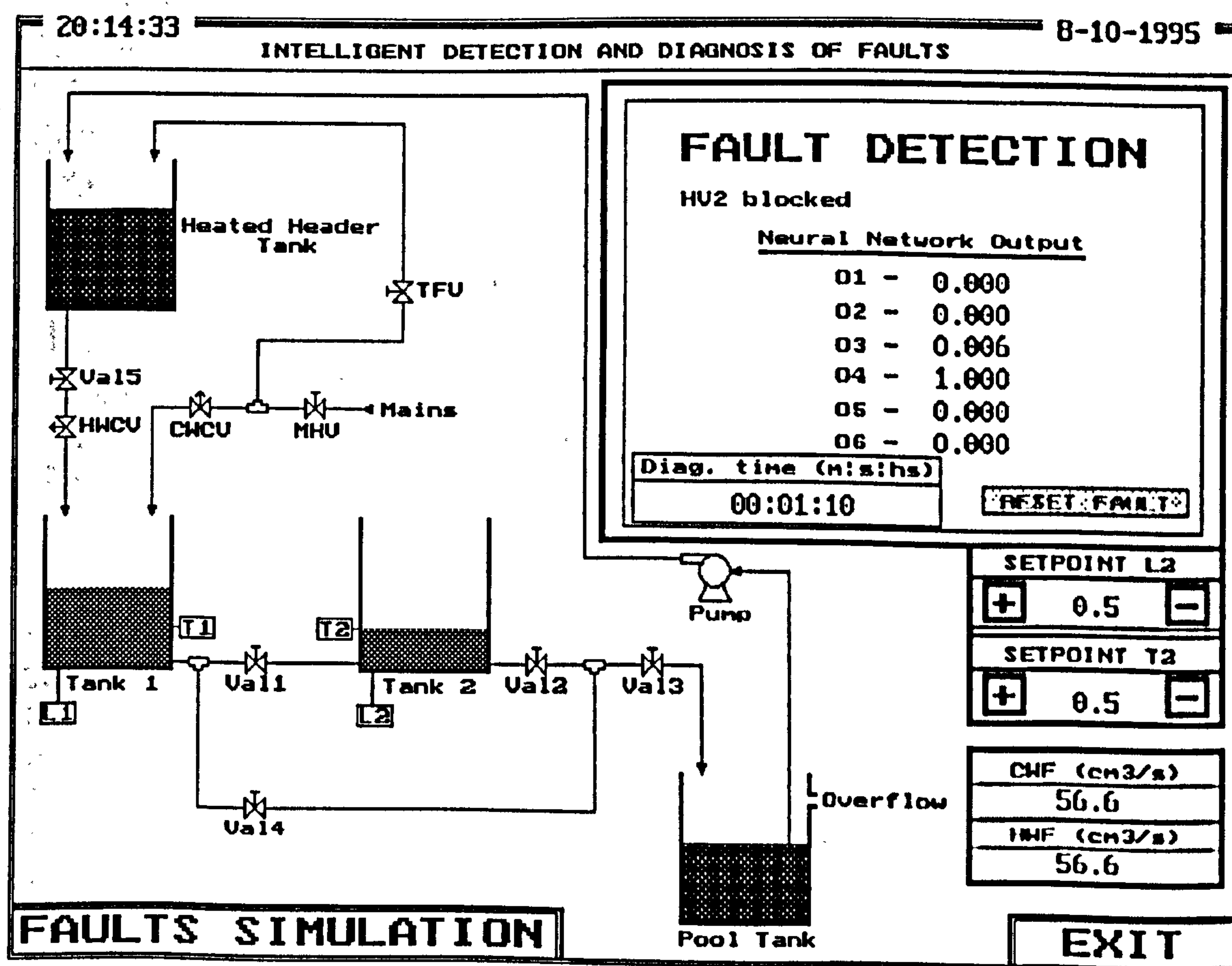


Figure 10.18 - Single abrupt fault detection in the mixing process



faults quoted above, which occur suddenly, have been simulated with different steady state conditions. For instance, Figure 10.18 shows the result achieved 1.1 second after the single abrupt fault "Hand valve 2 is blocked" has been simulated in the mixing process, with the setpoints of the controlled variables having the value 0.5. It can be seen that the diagnosis was very precise since the output 4 of the fuzzy neural network, which corresponds to the fault quoted above, has taken exactly the value 1 and all the other outputs have taken the value 0 or very close.

Figure 10.19 shows the result achieved in the mixing process under a double simultaneous abrupt fault situation. This diagnosis was achieved 1.1 seconds after the double simultaneous abrupt faults "Hand valve 2 is blocked and Cold water control valve fails low" have been simulated in the mixing process, with the steady state conditions corresponding to the controlled variables having the setpoints values at 0.5. Note that all the fault detection and diagnosis times quoted above, as well as all that will be mentioned below, have been obtained running the computational system implemented in a 386 PC (25 MHz) fitted with a mathematical co-processor.

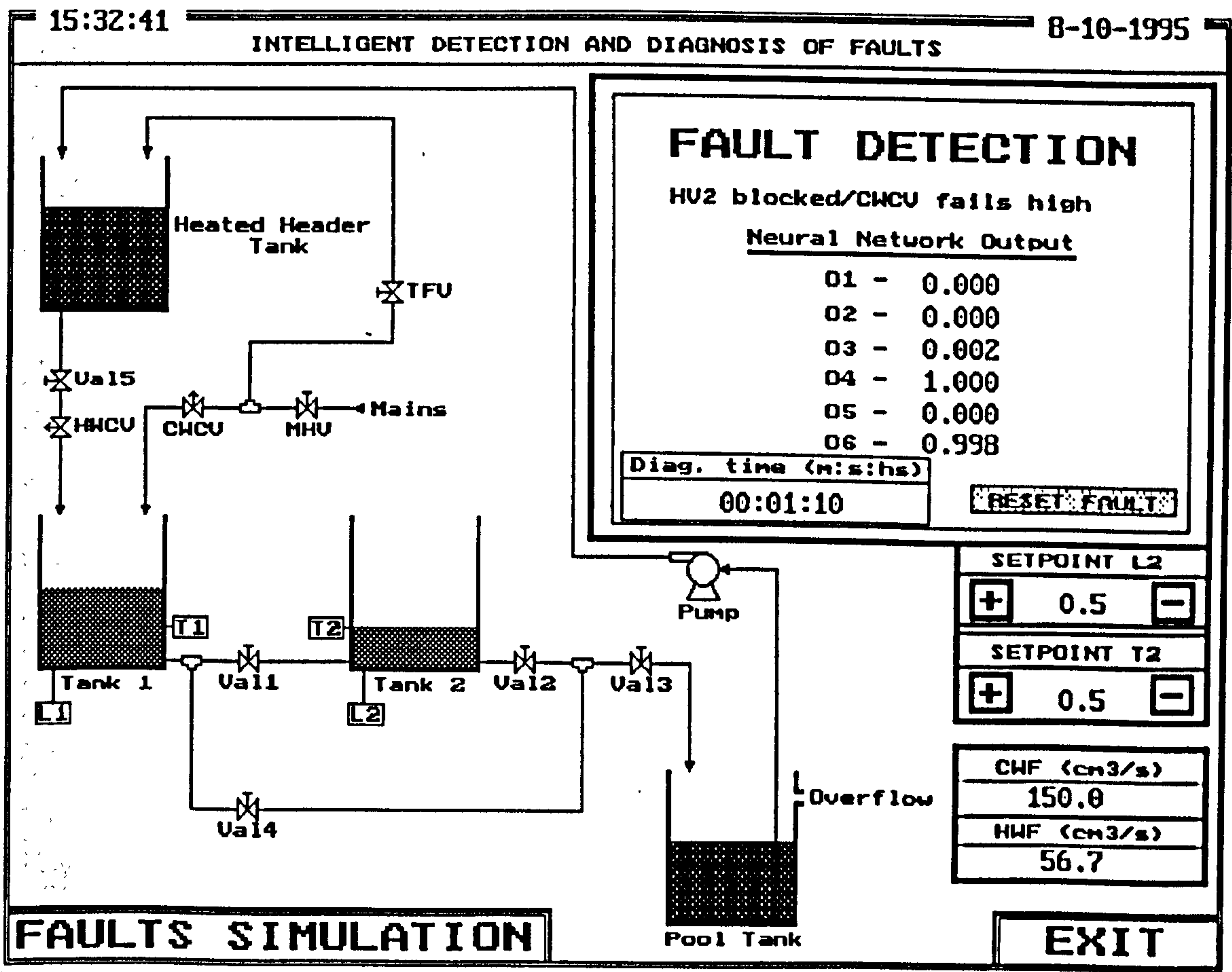


Figure 10.19 - Double simultaneous abrupt fault detection in the mixing process

All the single and double simultaneous faults, quoted above for both processes as abrupt faults, have also been considered during the current research work as incipient faults, which evolve

gradually. To simulate such an incipient faults in the process under concern, it has been assumed that the component degradation follows a linear law. Therefore, incipient faults are simulated through the following equation,

$$M_f = M_n \times (1 + \gamma \times t) \quad (10.18)$$

where the following notation is used,

$M_f$  - is the value of a process variable when there is a fault;

$M_n$  - is the value of a process variable when there is no fault;

$\gamma$  - is a constant which determines the speed of the fault development ( $sec^{-1}$ );

$t$  - is time ( $sec$ ).

In the remainder of this sub chapter some diagnostics are presented, which have been achieved for the mixing process under incipient failure situations. However, in order to make it possible to perform a comparison of the fault detection performance, under abrupt and incipient failure situations, the results shown below have been obtained under the same fault situations considered above.

Figure 10.20 shows the diagnosis achieved in the mixing process under a incipient single fault situation, "Hand valve 2 is blocked". This fault was simulated with a development speed of  $0.002 sec^{-1}$  and, as a result, the fault detection and diagnosis time has been 3.46 seconds against a fault detection and diagnosis time of 1.1 seconds obtained when the same fault was simulated as an abrupt fault. We can say that to achieve this result the generalisation ability of the fuzzy neural network has been used, since the training patterns used during the learning phase do not include incipient fault symptoms. Therefore, a small degradation in the fuzzy neural network outputs accuracy can be observed, but the system is still able to identify the correct fault.

Figure 10.21 shown a successful diagnostic which has been achieved under incipient double simultaneous faults situation. In this example, the double simultaneous faults, "Hand valve 2 is blocked and Cold water control valve fails low", have been simulated as incipient faults with a faults developing speed equal to  $0.02 sec^{-1}$ . From Figure 10.21, it can be seen that a successful diagnosis has been achieved 1.26 seconds after the faults have been initiated. Note that a fault detection and diagnosis time of 1.1 second was obtained when the same faults were simulated as abrupt faults. As in the diagnosis presented above for incipient single fault situations, when incipient double simultaneous faults are considered a small degradation in the fuzzy neural network outputs accuracy can be observed. Despite such a degradation, the fault detection and



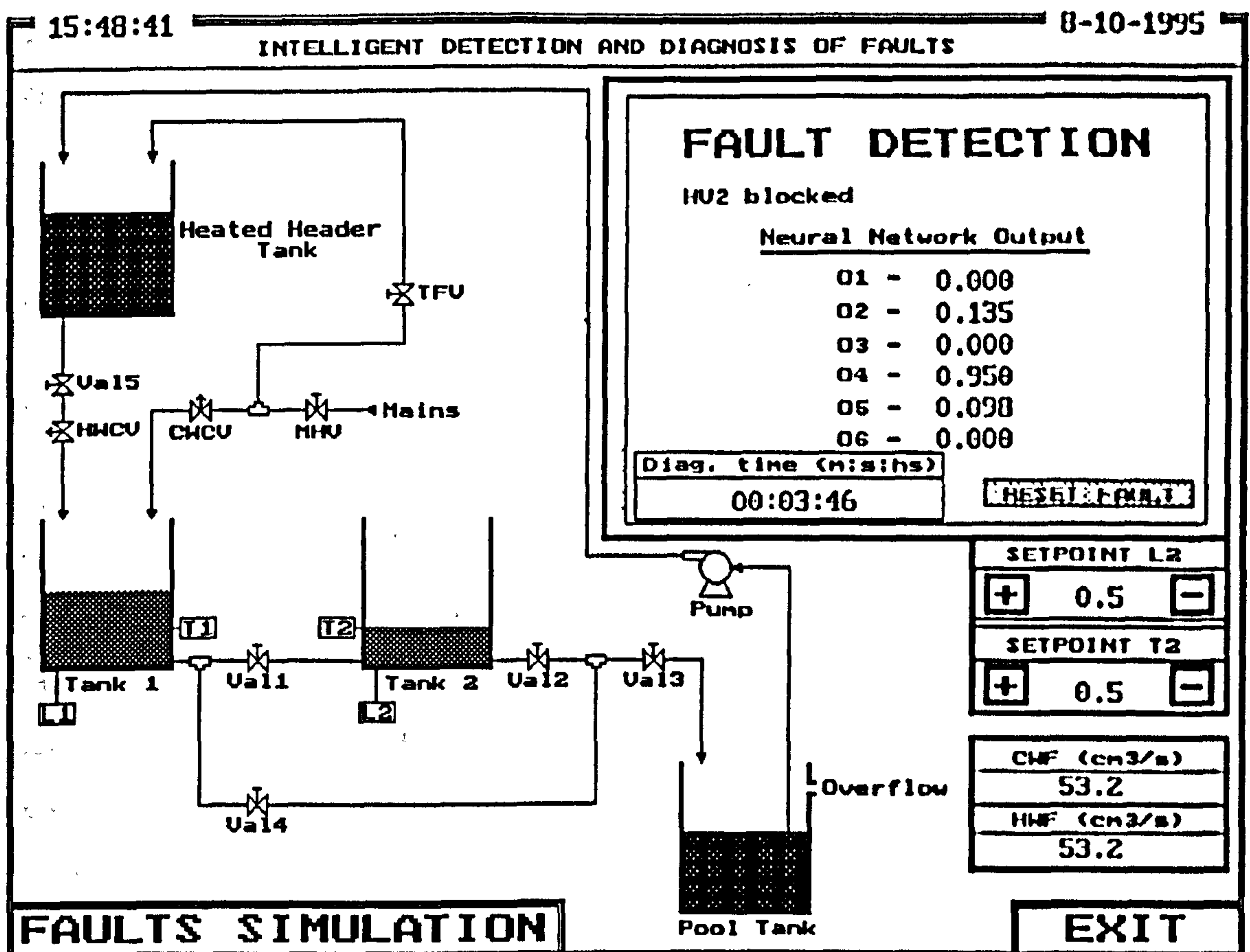


Figure 10.20 - Incipient single fault detection in the mixing process

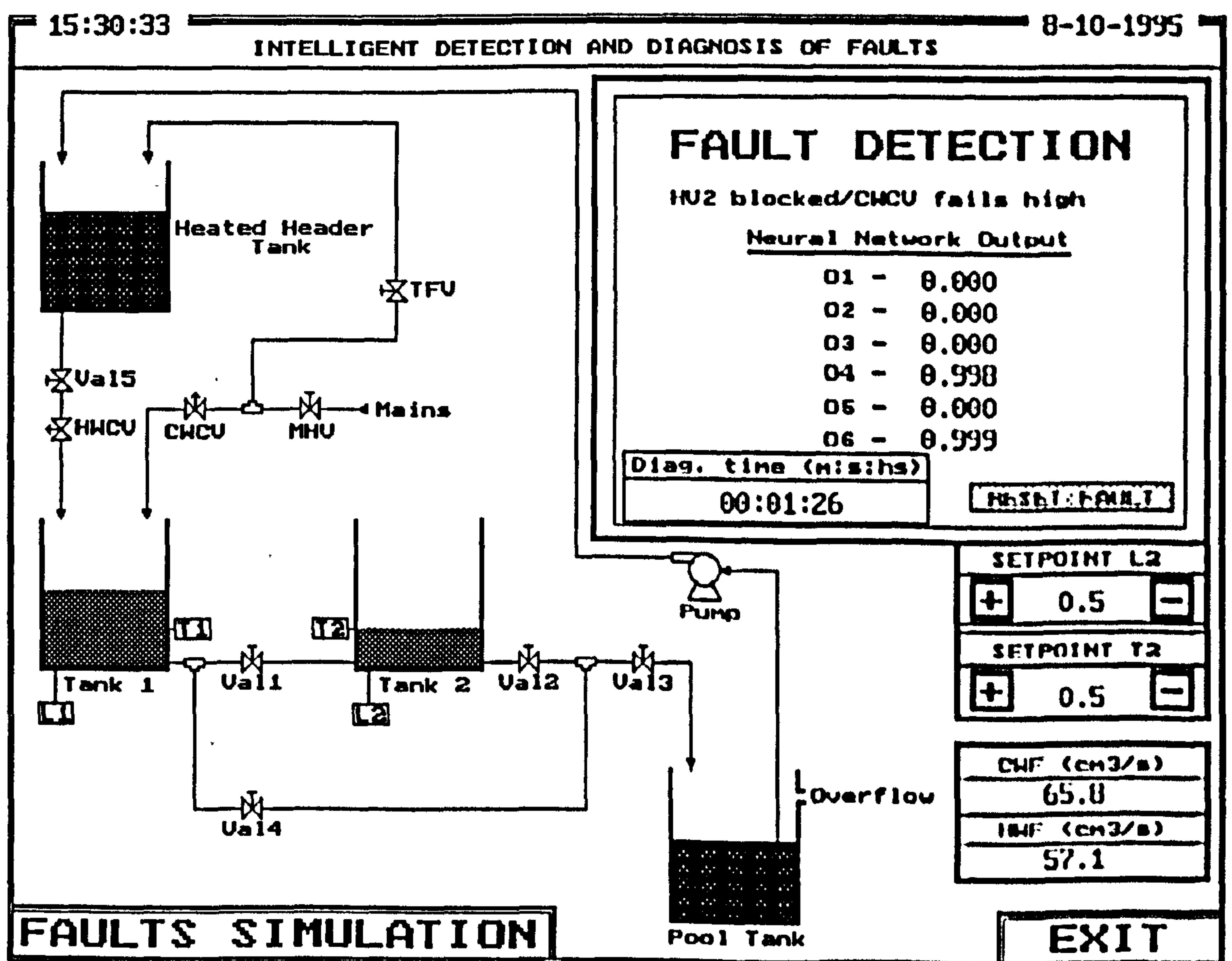


Figure 10.21 - Incipient double simultaneous fault detection in the mixing process

diagnosis system proposed in this chapter, is still able to detect and diagnose the correct double simultaneous faults.

## 10.7 Fault Detection and Diagnosis System Applied to the CSTR

As quoted in previous sub chapters, the continuous stirred tank reactor, which is presented in chapter 5., has been also used during simulation studies conducted to test the performance and reliability of the distributed intelligent fault detection and diagnosis approach proposed in this chapter. All the single and double simultaneous faults, which are represented in Tables 7.2 and 7.4 respectively, have been considered as abrupt and incipient faults. Incipient faults, which evolve gradually, have been simulated in an analogous way as described for the mixing process. Successful results have been obtained and are presented in the remainder of this sub chapter.

The fault detection heuristic rules have been generated in the last chapter while the fault diagnosis system is based on a fuzzy neural network with the characteristics previously presented. As quoted in sub chapter 10.5, nine measurement variables have been used as input data to the fuzzy neural network. These measurement variables are the following:

- $L$ , level in the reactor;
- $C_a$ , concentration of the reactant in the reactor;
- $C_b$ , concentration of the product in the reactor;
- $T$ , temperature in the reactor;
- $T_1$ , temperature of input reactant;
- $Q_1$ , flow rate of input reactant;
- $Q_2$ , flow rate of the recycled reactant;
- $Q_4$ , flow rate of the product;
- $Q_5$ , flow rate of the cold water entering the heat exchanger.

Thus, the diagnosis task is performed by presenting the changes in the measurement variables to the fuzzy neural network, which are propagated in a feed forward manner through the neural network. Then to locate a fault or faults in the process an analysis of the fuzzy neural network output values is carried out. However, for the same reasons given in the previous sub chapter for the mixing process, the changes in the measurement variables are considered in different time instants. This means that the input data for the fault diagnosis system is constituted by changes in measurement variables  $L, C_a, C_b$  and  $T$  observed when the fault diagnosis module is



triggered, together with the changes in the measurement variables  $T_1$ ,  $Q_1$ ,  $Q_2$ ,  $Q_4$  and  $Q_5$  observed at a previous sampling. However, it is worth noting that during the current studies it was found that the diagnosis of all the single faults, listed in Table 7.2, could be achieved by only using the measurement variables  $L$ ,  $C_a$ ,  $C_b$  and  $T$ .

The fuzzy neural network fuzzification layer has 63 processing elements arranged in 9 groups corresponding to the 9 on-line information sources, with each group containing 7 processing elements as described in previous sub chapters. As for the mixing process the number of hidden processing elements was fixed at 20. Moreover, as it can be seen in Table 7.2, 12 single faults have been considered and, hence, the output layer is constituted by 12 processing elements, each one corresponding to a fault.

Training data was obtained following a similar procedure as for the mixing process. The setpoint values of level and temperature in the reactor vessel were fixed at 0.5 and the setpoint of the recycled product flow rate took values between 0.1 and 1. Thus, all the single and double abrupt faults have been simulated with the different steady state conditions. In this manner, 105 patterns were obtained corresponding to the fault symptoms and the nominal operating conditions.

With such a fault detection and diagnosis system, several simulation studies have been performed. For instance, Figure 10.22 illustrates the diagnosis achieved 1.04 seconds after the

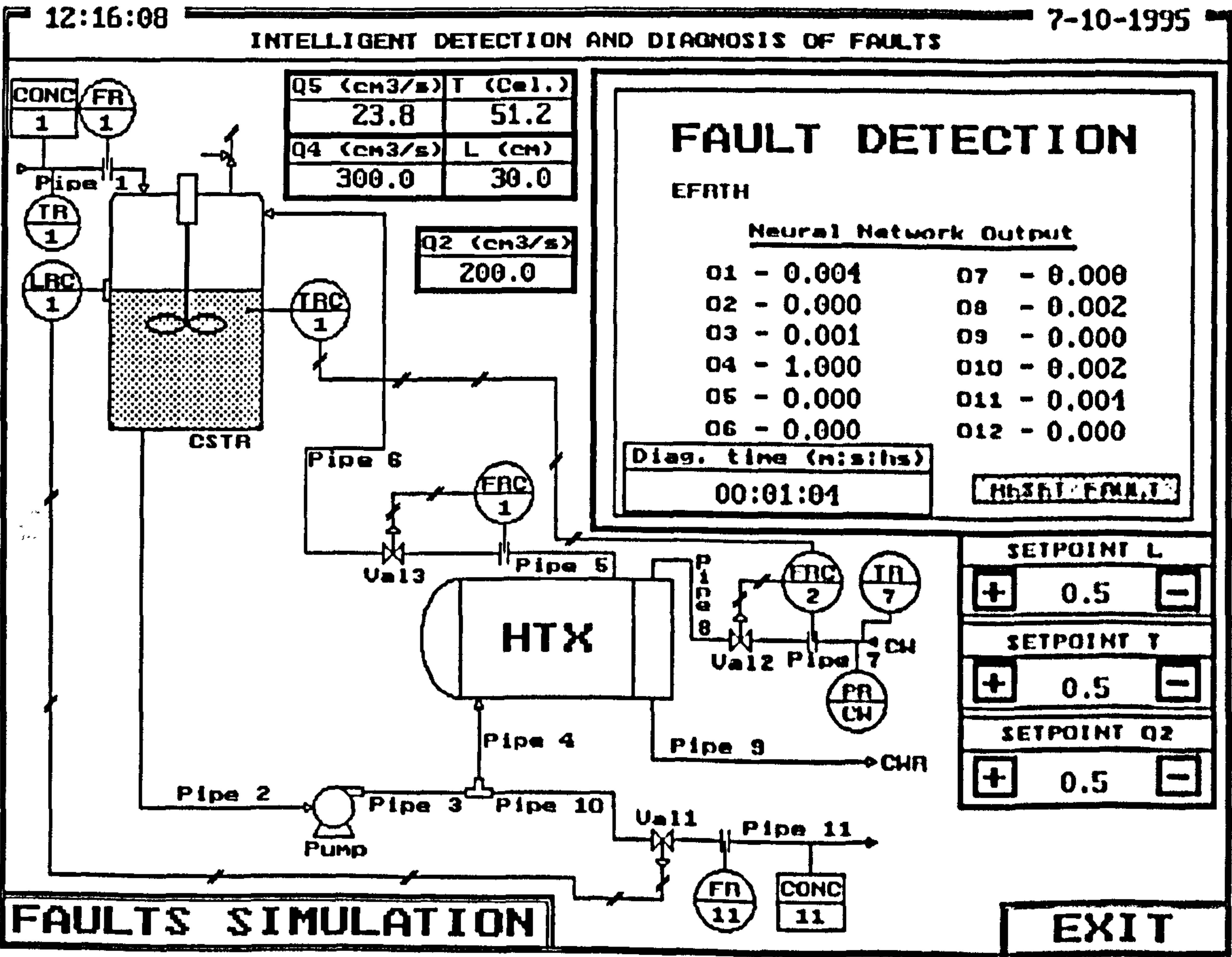


Figure 10.22 - Single abrupt fault detection in the CSTR



single fault "External feed reactant temperature high" has been initiated in the process, with all the setpoints of the controlled variables having the value 0.5. Once again, a very accurate diagnosis has been achieved, since the fuzzy neural network output corresponding to the fault quoted has taken the value 1 and the remaining outputs have taken the value 0 or very close.

Figure 10.23 shows the result achieved in the continuous stirred tank reactor under a double simultaneous abrupt fault situation. Here, the diagnosis was obtained 0.98 seconds after the double simultaneous abrupt faults "Pipe 2 or 3 is blocked or pump fails and External feed reactant temperature high" have been initiated. These faults have been simulated with the steady state conditions of the process corresponding to the setpoints of level and temperature into the reactor having the value 0.5 and the setpoint of the recycled flow rate having the value 0.6. A very accurate diagnosis is still observed.

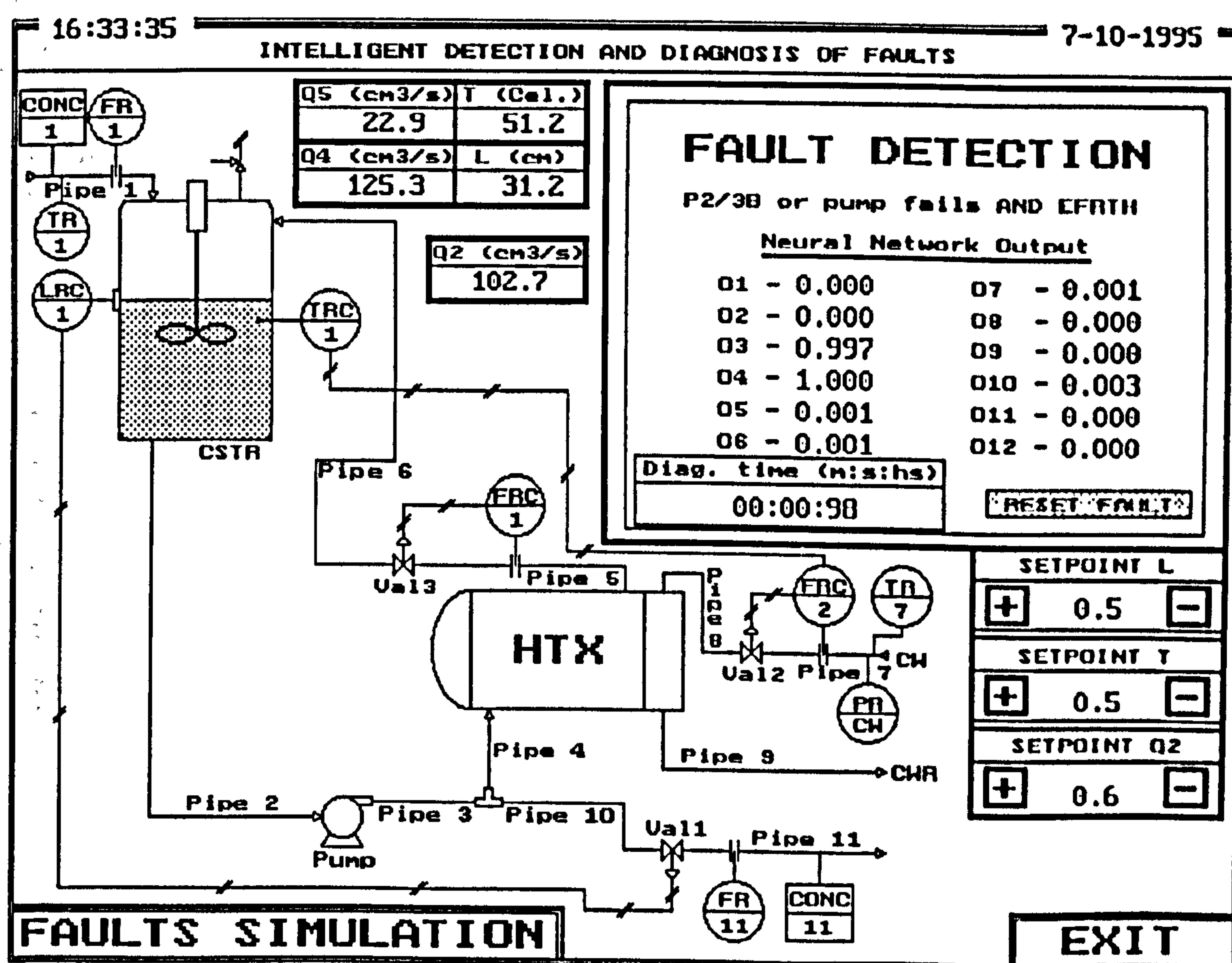


Figure 10.23 - Double simultaneous abrupt fault detection in the CSTR

In order to compare the fault detection and diagnosis system performance, under abrupt and incipient failure situations, both single and double simultaneous abrupt faults quoted above have been simulated as incipient faults under the same steady state conditions respectively. Therefore, Figure 10.24 shows the diagnosis result achieved in the continuous stirred tank reactor after the single fault, "External feed reactant temperature high", has been simulated as an incipient



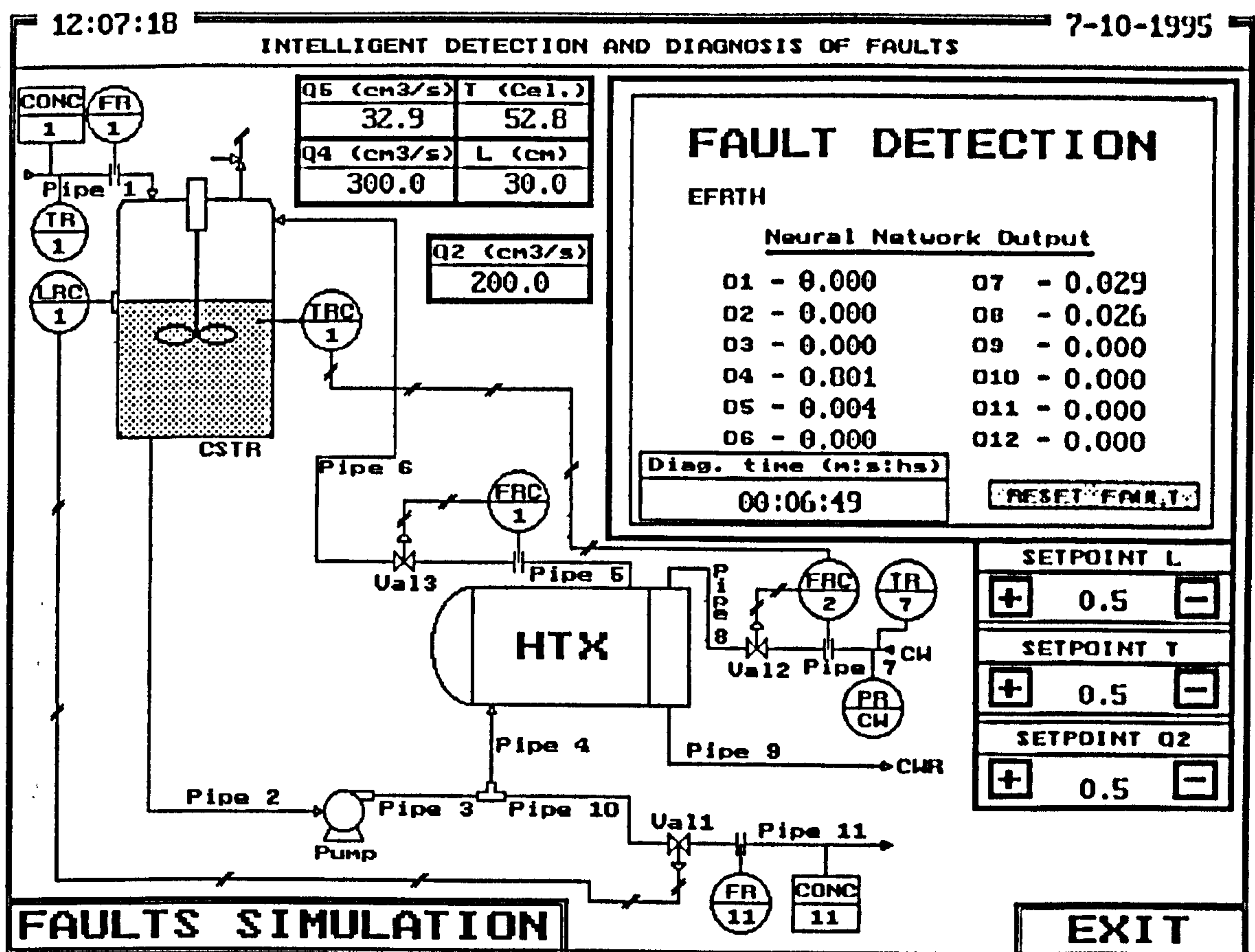


Figure 10.24 - Incipient single fault detection in the CSTR

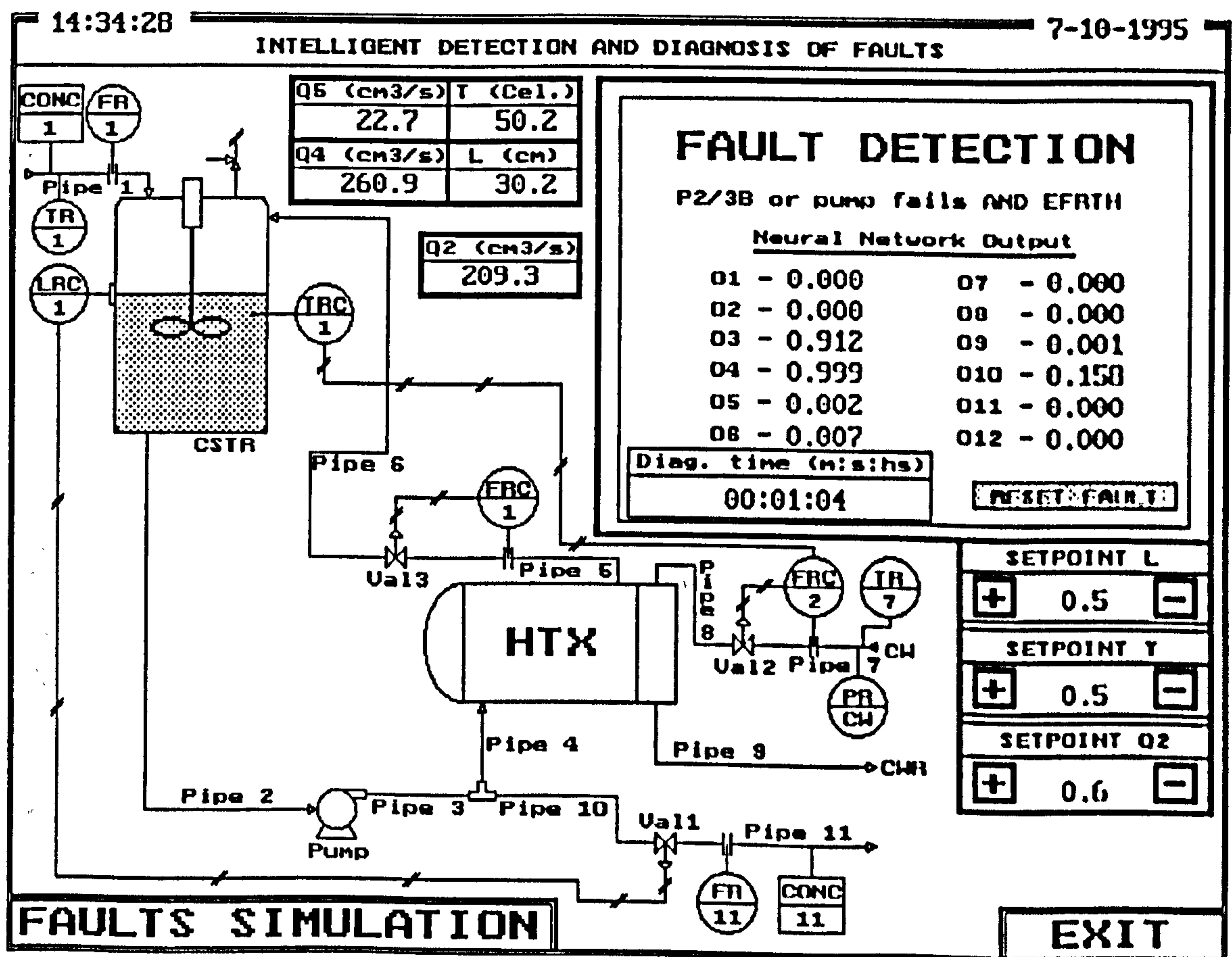


Figure 10.25 - Incipient double simultaneous fault detection in the CSTR



fault with a speed development value equal to  $0.0009 \text{ sec}^{-1}$ . It can be seen that after 6.49 seconds the fault has been successfully diagnosed. The result obtained under an incipient double simultaneous fault situation is depicted in Figure 10.25. Here, the double simultaneous faults, "Pipe 2 or 3 is blocked or pump fails and External feed reactant temperature high", have been considered under the steady state conditions quoted above for the simulation of such faults as abrupt. A fault developing speed of  $0.03 \text{ sec}^{-1}$  has been used, and the diagnosis has been achieved at 1.04 seconds after the faults have been simulated in the continuous stirred tank reactor.

For both cases, where incipient failure situations have been considered, a small degradation in the fuzzy neural network outputs accuracy can be observed. However, the overall distributed intelligent fault detection and diagnosis system proposed in this chapter is still able to detect and diagnose the correct faults occurred in the process. Some considerations about the performance of the system are presented in the next sub chapter.

## 10.8 Performance of the Fault Detection and Diagnosis System

As the results presented in previous sub chapters demonstrate, the overall distributed intelligent fault detection and diagnosis system has been successfully applied in simulation studies of the mixing process, as well as of a continuous stirred tank reactor. Good performance of the fault detection system has been observed avoiding false fault diagnosis under transient behaviours. For both processes a significative number of single and double simultaneous faults have been considered. All the faults quoted in the previous sections have been simulated as abrupt faults and successfully detected and diagnosed in time, with different steady state conditions of the process under consideration. Thus, we can state that the distributed fault detection and diagnosis system proposed in this chapter, has performed with good performance and high reliability has been achieved for the faulty scenarios.

Simulation studies under incipient fault or faults situations have also been conducted. Successful results have also been achieved under such faulty scenarios, as the diagnosis shown in the last two sub chapters demonstrates. However, a reliability degradation when the fault developing speed parameter take small values has been observed. This is due to the fact that the diagnosis of a incipient fault is not performed by matching the process variables behaviour, which are used as input data of the fuzzy neural network, with a pattern used during the learning procedure, but is performed using the neural network generalisation ability. Therefore, if under an incipient fault or faults situation the process variables behaviour is very different to the fault



symptoms used during the fuzzy neural network learning phase, such a fault or faults could be missed or misunderstood.

Nevertheless, in the author's opinion, this problem could be alleviated in practical implementations of the fault detection and diagnosis system, since the worst situation for the development of a incipient fault is to follow a linear law as considered in the present approach. Therefore, if the component degradation follows an exponentially increasing law, at a certain stage the process variables behaviour will be closer to the process variables behaviour under an abrupt fault situation and, hence, the problem quoted above will be minimised.

## 10.9 Conclusions

A distributed intelligent fault detection and diagnosis system, consisting of a knowledge based approach coupled with a fuzzy neural network, has been implemented. Successful results have been achieved during simulation studies conducted with a mixing process, as well as with a continuous stirred tank reactor. The system implemented has the ability to cope with transient behaviours of the process variables avoiding false fault detection and diagnosis under such situations.

The fault detection task has been performed through the knowledge based approach. Following the methodology proposed in the last chapter, fault detection heuristic rules, based on deep and shallow knowledge of the process under consideration, have been used to build up the knowledge base. The advantage of such a fault detection system is that the fault detection task is not dependent on threshold values and, hence, the performance and reliability of the overall fault detection and diagnosis system could be increased.

A fuzzy neural network approach has been proposed for fault diagnosis. The topology of the fuzzy neural network used to perform the fault diagnosis task is described in sub chapter 10.4. Moreover, an extension of the classical backpropagation supervised learning algorithm has been developed. It has been observed that this extension provides more efficient results than the same algorithm in its standard form. Several tests have been conducted which demonstrate that the extended algorithm provides a pre-defined accuracy in the learning results in less time than the standard algorithm. This suggests that if we are concerned with learning time the extension proposed for the standard backpropagation algorithm could be employed with advantage.

The fault diagnosis system based on a fuzzy neural network combines the advantages of both fuzzy reasoning and neural networks. Fuzzy reasoning is capable of handling uncertain and imprecise information while a neural network is capable of learning from examples. The fuzzy

neural network proposed for fault diagnosis purposes can be seen as a classical feed forward neural network with an additional fuzzification layer. Thus, quantitative information about the process being supervised is converted into qualitative information by the fuzzification layer. By using this qualitative approach to represent abnormalities in the process under consideration, similar training patterns are transformed into the same fault symptoms. Following this procedure training data is compressed and training effort is eased. Moreover, the use of qualitative information to perform the diagnosis task may reduce the sensitivity to measurement noise.

The successful results achieved with the distributed intelligent fault detection and diagnosis system suggest that the combined approach of a knowledge based system with a fuzzy neural network could be a powerful methodology for practical implementations. Following this methodology, transient behaviours of the process under concern will not affect the performance and reliability of the overall fault detection and diagnosis system.



## **Chapter 11**

# **Conclusions and Recommendations for Future Research**

The research carried out has been concerned with the application of artificial intelligence techniques to on-line process control and fault diagnosis, and the majority of this research is on on-line fault detection and diagnosis systems for industrial processes. Several on-line approaches have been developed and tested. The research results achieved with the implementation of a rule based controller demonstrate that this type of controller is useful in cases where mathematical models of the controlled process cannot be obtained or are very difficult to obtain, and/or in cases where the key control variables are not capable of being measured directly or reliably and, therefore, conventional control algorithms may not be efficiently applied. The research carried out with this type of controller, which is described in chapter 4., also suggests that the property of a rule based controller is mainly determined by the set of heuristic rules used on its conception and, hence, unlike conventional controllers, such as PI or PID controllers, the performance of a rule based controller is not as sensitive to its parameter changes.

On-line fault detection and diagnosis is regarded as a supervisory task in this research. Artificial intelligence techniques have more perspectives in performing such supervisory tasks than performing lower level regulatory tasks, since many supervisory tasks cannot be represented by a concise mathematical model.

Several different on-line fault detection and diagnosis systems for industrial processes have been investigated throughout this research. Two examples of an industrial plant, a mixing process and a continuous stirred tank reactor, have been used as test beds of various implemented on-line fault detection and diagnosis approaches. These processes are described in chapters 4. and 5.,



respectively, where the corresponding dynamic and qualitative models are derived. For both processes single and double simultaneous faults have been considered.

For fault detection and diagnosis purposes, knowledge based approaches, as well as fuzzy neural networks have been used. The research performed by using knowledge based systems emphasises the use of deep knowledge which can be qualitative models and/or knowledge on the connectivity and functions of process units. Two on-line fault detection and diagnosis approaches based on qualitative reasoning have been implemented and tested. The first one, which is presented in chapter 6., has been developed to cope with single abrupt faults that occur in the process under consideration. In chapter 7., an extension of the on-line fault detection and diagnosis system presented in chapter 6., which can cope with single and multiple simultaneous abrupt faulty scenarios, is considered.

In contrast with fault detection and diagnosis systems based on classical mathematical models, qualitative modelling provides a means for reasoning based on inaccurate process models and/or inaccurate measurements. Qualitative simulation is suitable for fault diagnosis purposes for which exact reasoning may not be necessary and, furthermore, since the exact severity of a fault is usually not known, qualitative simulation could be more appropriate for simulating the effect of a fault in the process under concern. In this context, there is motivation to avoid effort and expense of creating, maintaining and computing with rigorous dynamic mathematical models, by focusing on qualitative indicators of process conditions.

The simulation studies conducted with the on-line fault detection and diagnosis systems described in chapters 6. and 7. demonstrate that the confluence based qualitative reasoning technique (de Kleer and Brown 1984) is very suitable for detection of abrupt faults in the process under consideration. The set of confluence's for a process, which form the qualitative model, can be derived from its mathematical model. Using the confluence's representation, various fault models can be easily handled. The effect of a fault can be achieved by setting some variables in the qualitative model to certain specified qualitative values and, hence, it is not necessary to have different models for different operational conditions. To simulate the effect of a fault, it is only necessary to alter some variables in the qualitative model and all the operations based on the model remain unchanged. Therefore, it is not required to have different fault simulation procedures for different faults.

Ambiguity is a problem associated with qualitative reasoning. However, the research carried out on qualitative reasoning demonstrate that the use of fuzzy sets for representing the process variables behaviour result in a considerable reduction of the inherent ambiguity of qualitative computation. In contrast with the traditional qualitative simulation methods, the use of fuzzy sets allows a more detailed description of process variables through an arbitrary but finite



discretisation of the quantity space. The research performed with this fuzzy approach also suggests that qualitative reasoning ambiguity can be reduced or eliminated by using common sense knowledge. This knowledge can be incorporated in the basic descriptions of the quantity space through the use of graded membership within a fuzzy quantity space.

The work described in chapter 8. is based on, and supplements, the fuzzy qualitative modelling based fault detection and diagnosis approaches described in chapters 6. and 7.. The ability for reasoning about its own behaviour could make a knowledge based system more intelligent and autonomous and, therefore, it will be a desirable property of an on-line fault detection and diagnosis system. The on-line self-learning fault detection and diagnosis approach described in chapter 8. can be seen as a hierarchical system, where the lower level fault detection and diagnosis system is an ordinary one, identical to that described in chapters 6. and 7., and the upper level fault detection and diagnosis system will reason about the lower level one if any undesirable performance occurs there. By such means, any inappropriate parameters in the fault detection and diagnosis system can be found. Moreover, the research described in chapter 8. suggests that the use of machine learning techniques can be used very efficiently to automatically acquire fault symptoms in the form of heuristic rules and, hence, eases the knowledge acquisition task. Following this methodology, fault symptoms are acquired from on-line sampled data and, therefore, the diagnosis system will improve its performance in terms of diagnostic efficiency each time a new fault or faults occur in the process under consideration. This enhances diagnostic performance and reliability and can cover a wide range of potential faults.

The on-line fault detection and diagnosis systems based on fuzzy qualitative simulation have shown a good performance and reliability under single and multiple faulty scenarios. However, it has been observed a performance and reliability degradation when incipient faults are considered. This is mainly due to the fact that in such approaches the fuzzy qualitative simulation is only triggered when pre-defined threshold values are reached. Therefore, since for avoiding false alarms under transient behaviours of the process these threshold values can not be too small, the research conducted with the above mentioned fault detection and diagnosis approaches has shown that they are not very appropriate for diagnosing faults that evolve gradually, that is incipient faults. This problem has been investigated and, a distributed intelligent on-line fault detection and diagnosis system has been implemented for coping with incipient faults. This approach is based on a knowledge based system coupled with a fuzzy neural network. Fault detection is performed by the knowledge based system and, when a hypothetical fault or faults are detected the fuzzy neural network approach is triggered in order to locate the fault or faults in the process under concern. During the simulation studies carried out with the distributed on-line fault detection and diagnosis



system a good performance and reliability have been observed even under transient behaviours of the process under consideration.

The reasons mentioned above have motivated the research work described in chapters 9. and 10.. Chapter 9. suggests that developing fault detection heuristic rules based on knowledge on system structures and component functions would be a systematic way for developing rule based fault detection systems. Fault detection heuristic rules developed in such a way could cover a wide range of potential faults. Inference based on these heuristic rules has a higher certainty since these rules capture the underlying first principles of the process under consideration. Any experimental knowledge, that is shallow knowledge, can also be integrated with the heuristic rules developed from knowledge of system structures and component functions, that is deep knowledge. Moreover, since the methodology developed is systematic and the structural decomposition corresponds to plant topology, it may be appropriate for generating fault detection heuristic rules of large scale processes.

The fault detection knowledge based approach has been coupled with a fault diagnosis system based on a fuzzy neural network. This implementation avoids the use of threshold values for firing the overall on-line fault detection and diagnosis system and enables it to cope with transient behaviours of the process being diagnosed. Chapter 10. presents a fault diagnosis system based on a fuzzy neural network. This approach combines the capability of fuzzy reasoning in handling uncertain information and the capability of artificial neural networks in learning from examples. As a matter of fact, an advantage of such systems is that they are easy to develop provided that training data is available. Training data could be obtained from past operating experience on a process or from simulation studies. However, since in artificial neural networks, training time is usually a concern, an extension of the classical backpropagation learning algorithm has been developed. The research results presented in chapter 10. show the power of the extended algorithm for speeding up the fuzzy neural network training task. This aspect is particularly important when the complexity of the neural network increases and/or a huge number of training patterns are considered.

The research results presented in chapter 10. suggest that fuzzy neural network based diagnosis systems could work under partially incorrect information and, hence, they can tolerate model plant mismatch in the case where training data is obtained from simulation studies. This neural network ability has also been found to very useful for performing incipient faults diagnosis, since the training examples used only included abrupt faults symptoms. This demonstrates the robustness of fuzzy neural network based diagnosis systems. A further advantage of the fuzzy neural network based approach is the parallel nature in neural network operations, which can be ideally implemented with the recently developed parallel processing techniques to achieve real time



requirements. From the practical point of view, the use of fuzzy sets together with artificial neural networks may alleviate the problem usually associated with the effects of measurement noise.

The research results presented in this thesis have shown the great potential of artificial intelligence techniques in performing on-line process control tasks including both lower level regulation tasks and higher level supervisory tasks. The on-line fault detection and diagnosis systems described in this thesis have been successfully applied in simulation studies conducted with a mixing process and with a continuous stirred tank reactor. Single and double simultaneous faults have been considered, together with abrupt and incipient faults. Further applications to industrial scale processes could be investigated in future research. Since the on-line fault detection and diagnosis systems developed in this research will not have any side effects on the process being monitored, they are ready for industrial trial.

As quoted in chapter 6., when applying fuzzy systems in practice, the main topic for the designer is to find a good parameter set of membership functions describing the linguistic terms in order to achieve the desired results. The suitable membership functions have usually been given by a very time consuming trial and error procedure. In general, following this procedure we can not be sure that the selected membership functions will provide the system with a better performance. Furthermore, it is desirable for control engineers to tune the parameters in order to achieve an optimal system performance. Therefore, the recently emerged topic in artificial intelligence, Genetic Algorithms (Goldberg 1989, Koza 1994), which imitates the process of biological evolution and has shown remarkable performance in search, optimisation and machine learning, has potential perspectives for selecting the most appropriate membership functions for a specific task based on some performance measures. Genetic algorithms are search procedures based on natural selection and natural genetics and are efficient for global searches. Generally, genetic algorithms consist of three operators: reproduction, crossover and mutation. Given a search problem, genetic algorithms run repeatedly by iteration by using the three operators at random, but based on a fitness function evolution to find a better solution in the searching space. Genetic algorithms require only information concerning the quality of the solution (the fitness value) produced by each parameter set. This differs from many searching methods requiring derivative information or complete knowledge of the problem structure. Due to the simple structure of genetic algorithms and since this structure does not rely on the characteristics of the system being considered, genetic algorithms may be applicable to a large range of practical problems. Further research is needed to explore the perspectives of genetic algorithms incorporated into on-line fault detection and diagnosis systems for industrial processes, in order to make them more intelligent and autonomous.

As mentioned above, a problem associated with knowledge based fault detection and diagnosis systems is the detection and diagnosis of incipient faults. With the goal to overcome this



problem and, hence, increase the performance and reliability of these systems, the combination of a knowledge based approach with a parameter estimation approach could be a future topic of research. Under such a scheme, parameter estimation techniques could function as part of an information pre-processor, which provides the knowledge based element with more information about the process under consideration. Such a system could then be sensitive to slight faults.

Process supervisory tasks include on-line fault detection and diagnosis and other tasks such as suggesting repairing procedures after a fault has been diagnosed, suggesting different control structures and control algorithms in cases of occurrences of faults. These tasks are specially important since a conventional feedback control loop design for a large scale system may result in unsatisfactory performance, or even instability, in the event of malfunctions in process components. A closed loop control system which tolerates component malfunctions, whilst still maintaining desirable performance and stability properties can be said to be a fault tolerant control system which has attracted the attention of several researchers (Patton 1993, Stengel 1993). The conventional approach to fault tolerant control systems includes the design of three separate modules: control, fault detection and diagnosis, and reconfiguration. The independent design between the control and fault detection and diagnosis modules may neglect the significant interactions between them and, hence, the reconfiguration module may fail to maintain the desirable stability and performance of the system. There is, therefore, a need for a research study into this subject where artificial intelligence techniques could play a major role.

Many on-line fault detection and diagnosis approaches have been developed. However many techniques are very complicated to apply without the assistance of design software. Hence, there is a need for developing a design toolbox which can be used for new applications and further research. This toolbox should of course have a modular structure and a common information exchange standard between modules. The user would be able to select the most appropriate diagnostic technique to suit a particular problem. Moreover, the user should be able to combine different modules in order to form a complete application for a given situation, and the toolbox should provide the most efficient way for linking and assuring data communication between units.



# References

- Altrock, C. von (1995), "Fuzzy Logic Applications in Europe", in J. Yen, R. Langari and L. A. Zadeh (eds.), *Industrial Applications of Fuzzy Logic and Intelligent Systems*, IEEE Press, pp277-310.
- Antsaklis, P. and K. Passino (1992), *An Introduction to Intelligent and Autonomous Control*, Kluwer.
- Asse, A., A. Maizener, A. Moreau and D. Willaeyts (1987), "Diagnosis Based on Subjective Information in a Solar Energy Plant", in E. Sanchez and L. A. Zadeh (eds.), *Approximate Reasoning in Intelligent Systems, Decision and Control*, Pergamon Press, Oxford, pp159-173.
- Bellman, R., M. Kalaba and L. A. Zadeh (1964), "Fuzzy Sets", *Information and Control*.
- Bezdek, J. C. and S. K. Pal (eds.) (1992), *Fuzzy Models for Pattern Recognition*, IEEE Press.
- Blum, A. (ed.) (1992), *Neural Networks in C++ - An Object-Oriented Framework for Building Connectionist Systems*, John Wiley & Sons Inc.
- Bonissone, P. P. and K. S. Decker (1986), "Selecting uncertainty calculi and granularity: An experiment in trading-off precision and complexity", in L. N. Kanal and J. F. Lemmer (eds.), *Uncertainty in Artificial Intelligence*, Amsterdam: North-Holland, pp217-247.
- Buckley, J. J. and Y. Hayashi (1994), "Fuzzy Neural Networks", in R. R. Yager and L. A. Zadeh (eds.), *Fuzzy Sets, Neural Networks, and Soft Computing*, Van Nostrand Reinhold, New York, pp233-249.
- Campos, L. M. and S. Moral (1993), "Learning rules for a fuzzy inference model", *Fuzzy Sets and Systems*, Vol. 59, pp247-257.
- Charniak, E. and D. McDermott (eds.) (1985), *Introduction to Artificial Intelligence*, Addison-Wesley.

- Chen, C.-L., P.-C. Chen and C.-K. Chen (1993), "Analysis and design of fuzzy control systems", *Fuzzy Sets and Systems*, Vol. 57, pp125-140.
- Chen, J. (1995), *Robust Residual Generation for Model-Based Fault Diagnosis of Dynamic Systems*, PhD Thesis, Department of Electronics, University of York, UK.
- Clancey, W. J. (1985), "Heuristic classification", *Artificial Intelligence*, Vol. 27, pp289-350.
- Dayhoff, J. (1990) (ed.), *Neural Networks Architectures*, Van Nostrand Reinhold.
- de Kleer, J. and J. S. Brown (1984), "A Qualitative Physics Based on Confluences", *Artificial Intelligence*, Vol. 24, pp7-83.
- de Kleer, J. and B. C. Williams (1987), "Diagnosing Multiple Faults", *Artificial Intelligence*, Vol. 32, pp97-130.
- Denning, P. J. (1986), "Towards a Science of Expert Systems", *IEEE Experts*, Vol. 1, No.2, pp80-83.
- Donald, J. H. (1994), "Rule induction - machine learning techniques", *Computing & Control Engineering Journal*, Vol. 5, N<sup>o</sup>. 5, pp249-255.
- Dubois, D. and H. Prade (eds.) (1980), *Fuzzy set and Systems: Theory and Applications*, Academic Press, New York.
- Ellis, J. E., D. S. Wadwani, C. Maxwell and S. N. Das (1986), "Supervisory Direct Digital Control of a Mixing Process", *The City University Research Memorandum*, CEC/JEE-DSW-CM-SND/42.
- Fahlman, S. E. (1988), "Faster-Learning Variations on Back-Propagation: An empirical study", in T. L. Sejnowski, G. E. Hinton and D. S. Touretzky (eds.), *Proceedings of 1988 Connectionist Models Summer School*, CA: Morgan Kaufmann, pp38-51.
- Finch, F. E. and M. A. Kramer (1988), "Narrowing diagnostic focus using functional decomposition", *AIChE Journal*, Vol. 34, N<sup>o</sup>. 1, pp25-36.



- Forbus, K. D. (1984), "Qualitative Process Theory", *Artificial Intelligence*, Vol. 24, pp85-168.
- Forsyth, R. and R. Rada (1986), *Machine Learning: Applications in Expert Systems and Information Retrieval*, Ellis Horwood Limited.
- Frank, P. M. (1990), "Fault Detection in Dynamic Systems Using Analytical and Knowledge-Based Redundancy - A Survey and Some New Results", *Automatica*, Vol. 26, No. 3, pp450-472.
- Franks, R. G. E. (ed.) (1972), *Modeling and Simulation in Chemical Engineering*, John Wiley & Sons, Inc..
- Genesereth, M. R. (1984), "The use of design descriptions in automated diagnosis", *Artificial Intelligence*, Vol. 24, pp411-436.
- Goldberg, D. E. (ed.) (1989), *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley Publishing Company, Inc..
- Guan, J. and J. H. Graham (1994), "Diagnostic Reasoning With Fault Propagation Digraph and Sequential Testing", *IEEE Trans. Systems, Man and Cybernetics*, Vol. 24, N°. 10, pp1552-1558.
- Harmon, P. and D. King (eds.) (1985), *Expert Systems*, John Wiley & Sons, Inc.
- Haykin, S. (ed.) (1994), *Neural Networks - A Comprehensive Foundation*, Macmillan College Publishing Company, Inc.
- He, S.-Z., S. Tan, F.-L. Xu and P.-Z. Wang (1993), "Fuzzy self-tuning of PID controllers" *Fuzzy Sets and Systems*, Vol. 56, pp37-46.
- Hebb, D. O. (ed.) (1949), *The Organization of Behavior*, John Wiley & Sons.
- Herbert, M. R. and G. H. Williams (1987), "An Initial Evaluation of the Detection and Diagnosis of Power Plant Faults Using a Deep Knowledge Representation of Physical Behaviour", *Expert Systems*, Vol. 4, No. 2, pp90-99.

- Himmelblau, D. M., R. W. Barker and W. Suewatanakul (1991), "Fault classification with the aid of artificial neural networks", *Preprints of IFAC/IMACS Symposium SAFEPROCESS'91*, Baden, Vol. 2, pp369-373.
- Hopfield, J. J. (1982), "Neural Networks and Physical Systems with Emergent Collective Computational Abilities", *Proc. National Academy of Science (USA)*, Vol. 79, pp2554-2558.
- Hopfield, J. J. and D. W. Tank (1986), "Computing with Neural Circuits: A Model", *Science*, Vol. 233, pp625-633.
- Hornik, K., M. Stinehcombe and H. White (1989), "Multilayer feedforward networks are universal approximators", *Neural Networks*, Vol. 2, pp359-366.
- Hoskins, J. C., K. M. Kaliyur and D. M. Himmelblau (1991), "Fault Diagnosis in Complex Chemical Plants", *AIChE Journal*, Vol. 37, No. 1, pp137-141.
- Iri, M., K. Aoki, E. O'Shima and H. Matsuyama (1979), "An Algorithm for Diagnosis of System Failures in the Chemical Process", *Computers and Chemical Engineering*, Vol. 3, pp489-493.
- Isermann, R. (1984), "Process Fault Detection Based on Modeling and Estimation Methods - A Survey", *Automatica*, Vol. 20, No. 4, pp387-404.
- Jackson, P. (ed.) (1986), *Introduction to Expert Systems*, Addison-Wesley.
- Jackson, P. (ed.) (1990), *Introduction to Expert Systems*, 2nd Edition, Addison-Wesley.
- Jang, J.-S. R. (1992), "Self-Learning Fuzzy Controllers Based on Temporal Back Propagation", *IEEE Transactions on Neural Networks*, Vol. 3, No. 5, pp714-723.
- Johnson, L. and E. T. Keravnou (eds.) (1984), *Expert Systems Technology: A Guide*, Abacus Press.



- Kahn, G. (1988), "MORE: from observing knowledge engineers to automating knowledge acquisition", in S. Marcus (ed.), *Automating Knowledge Acquisition for Expert Systems*, Boston: Kluwer Academic, pp7-35.
- Kandel, A. and S. C. Lee (eds.) (1979), *Fuzzy Switching and Automata - Theory and Applications*, Computer Systems Engineering Series, Russak & Company Inc..
- Kavuri, S. N. and V. Venkatasubramanian (1994), "Neural network decomposition strategies for large-scale fault diagnosis", *Int. Journal Control*, Vol. 59, No. 3, pp767-792.
- Kitowski, J. and M. Bargiel (1988), "Diagnostics of faulty states in complex physical systems using fuzzy relational equations", in E. Sanchez and L. A. Zadeh (eds.), *Approximate Reasoning in Intelligent Systems, Decision and Control*, Pergamon Press, Oxford, pp175-194.
- Klir, J. G. and T. A. Folger (eds.) (1988), *Fuzzy Sets, Uncertainty, and Information*, Prentice Hall.
- Kosko, B. (ed.) (1992), *Neural Networks and Fuzzy Systems - A Dynamical Systems Approach to Machine Intelligence*, Prentice-Hall International Editions.
- Koza, J. R. (ed.) (1994), *Genetic Programming II, Automatic Discovery of Reusable Programs*, The MIT Press.
- Kramer, M. A. and B. L. Palowitch (1987), "A Rule-Based Approach to Fault Diagnosis Using the Signed Directed Graph", *AIChE Journal*, Vol. 33, No. 7, pp1067-1078.
- Kramer, M. A. and F. E. Finch (1989), "Fault diagnosis of chemical process", in S. G. Tzafestas (ed.), *Knowledge-based system diagnosis, supervision and control*, Plenum Press, pp247-263.
- Kuipers, B. (1986), "Qualitative Simulation", *Artificial Intelligence*, Vol. 29, pp289-338.
- Kuipers, B. (1987), "Qualitative Simulation as Causal Explanation", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 17, No. 3, pp432-443.

Langari, R., J. Yen and J. Painter (eds.) (1994), *Proceedings of the Third International Conference on Industrial Applications of Fuzzy Control and Intelligent Systems*, Houston, Texas, December 2-4, IEEE Press.

Lee, C. C. (1992), "Fuzzy logic in control systems: Fuzzy logic controller - Part I, Part II", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 20, pp404-418, pp429-435.

Leonard, J. A. and M. A. Kramer (1990), "Improvement of the back-propagation algorithm for training neural networks", *Computers & Chemical Engineering*, Vol. 14, pp337-341.

Linkens, D. A. and H. O. Nyongesa (1995), "Genetic algorithms for fuzzy control, Part 1: Offline system development and application, Part 2: Online system development and application", *IEE Proc.-Control Theory Applications*, Vol. 142, No. 3, pp161-176, pp177-185.

Lippmann, R. P. (1987), "An Introduction to Computing with Neural Nets", *IEEE ASSP Magazine*, April, pp4-22.

Marshall, G. (ed.) (1990), *Advanced Students' Guide to Expert Systems*, Heinemann Newnes.

McCulloch, W. S. and W. Pitts (1943), "A Logical Calculus of the Ideas Imminnent in Nervous Activity", *Bulletin of Mathematical Biophysics*, No. 5, pp115-133.

McDowell, J. K., M. A. Kramer and J. F. Davis (1991), "Knowledge-Based Diagnosis in Proccss Engineering", *IEEE Expert*, June, pp65-66.

Michalski, R. S., J. G. Carbonell and T. M. Mitchell (eds.) (1983), *Machine Learning: An Artificial Intelligence Approach*, Tioga, CA: Palo Alto.

Michalski, R. S., J. G. Carbonell and T. M. Mitchell (eds.) (1986), *Machine Learning: An Artificial Intelligence Approach*, Vol. II, Morgan Kaufmann Publishers Inc..

Michalski, R. S. and R. L. Chilansky (1980), "Learning by Being Told and Learning from Examples: an Experimental Comparison of the two Methods of Knowledge Acquisition in the Context of Developing an Expert System for Soybean



Disease Diagnosis", *Policy Analysis and Information Systems*, Vol. 4, No. 2, pp125-160.

Michie, D. (1982), "The State of the Art in Machine Learning", in D. Michie (ed.), *Introductory Readings in Expert Systems*, Gordon and Breach, London, pp208-229.

Minsky, M. L. and S. A. Papert (eds.) (1969), *Perceptrons*, The MIT Press.

Mitchell, T. M., J. G. Carbonell and R. S. Michalski (eds.) (1986), *Machine Learning: A Guide to Current Research*, Kluwer Academic Publishers.

Mizumoto, M. and K. Tanaka (1978), "Some Properties of Fuzzy Sets under Various Operations", in J. Rose (ed.), *Current Topics in Cybernetics and Systems* (Proceedings of the Fourth International Congress of Cybernetics and Systems, Amsterdam, August 21-25, 1978), World Organisation of General Systems and Cybernetics and Spring Verlag, Berlin, pp371-372.

Montague, G. A., A. J. Morris and M. J. Willis (1992), "Artificial neural networks methodologies and applications in process control", in K. Warwick, G. W. Irwin and K. J. Hunt (eds.), *Neural Networks for control and systems*, Peter Peregrinus Ltd on behalf of the IEE, pp31-50.

Moor, R. L. and M. A. Kramer (1986), "Expert Systems in On-Line Process Control", *Chemical Control III*, Asilomer, pp839-867.

Naidu, S., E. Zafiriou and T. J. McAvoy (1990), "Use of neural networks for failure detection in a control system", *IEEE Control Syst. Mag.*, Vol. 10, pp49-55.

Narayanan, N. H. and N. Viswanadham (1987), "A Methodology for Knowledge Acquisition and Reasoning in Failure Analysis of Systems", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-17, No. 2, pp274-288.

Nilsson, N. J. (ed.) (1980), *Principles of Artificial Intelligence*, Palo Alto, CA: Morgan Kaufmann.

- Oyeleye, O. O., M. A. Kramer (1988), "Qualitative Simulation of chemical Process Systems: Steady-State Analyses", *AIChE Journal*, Vol. 34, No. 9, pp1441-1453.
- Padalkar, S., G. Karsai, C. Biegl, J. Sztipanovits, K. Okuda and N. Miyasaka (1991), "Real-Time Fault Diagnostics", *IEEE Expert*, June, pp75-85.
- Pal, S. K. and D. K. D. Majumder (eds.) (1986), *Fuzzy Mathematical Approach to Pattern Recognition*, Wiley Eastern Limited.
- Park, J. and I. W. Sanberg (1991), " Universal approximation using radial basis function networks", *Neur. Comput.*, Vol. 3, pp246-257.
- Patton, R. J. (1993), "Robustness issues in fault tolerant control", *Proceedings of International Conference on Fault Diagnosis: TOOLDIAG'93*, Toulouse, France, Plenary Paper.
- Patton, R. J., J. Chen and T. Siew (1994), "Fault Diagnosis in Nonlinear Dynamic System via Neural Networks", *Proceedings of UKACC/IEE International Conference CONTROL'94*, March 21-24, pp1346-1351.
- Patton, R. J., P. M. Frank and R. N. Clark (eds.) (1989), *Fault Diagnosis in Dynamic Systems - Theory and Applications*, Prentice Hall.
- Pazzani, M. J. (1986), "Refining the Knowledge Base of a Diagnosis Expert System: An Application of Failure-Driven Learning", *Proceedings of AAAI-86*, Philadelphia, August 11-15, pp1029-1035.
- Pazzani, M. J. (1987), "Failure-Driven Learning of Fault Diagnosis Heuristics", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-17, No. 3, pp380-394.
- Peng, X. T. and S. M. Lin (1988), "Self-learning PID controllers and its applications to a temperature controlling process", in M. Gupta and T. Yamakawa (eds.), *Fuzzy Computing*, North Holland, Amsterdam, pp355-363.
- Pham, D. T. (ed.) (1988), *Expert Systems in Engineering, Artificial Intelligence in Industry*, IFS Publications Ltd..



Price, C. and M. Lee (1988), "Application of Deep Knowledge", *Artificial Intelligence in Engineering*, Vol. 3, No. 1, pp12-17.

Raiman, O. (1986), "Order of Magnitude Reasoning", *Proceedings of AAAI-86*, Philadelphia, August 11-15, pp100-104.

Rao, N. S. V. and N. Viswanadham (1987), "Fault diagnosis in dynamic systems: a graph theoretic approach", *International Journal Systems Science*, Vol. 18, No. 4, pp687-695.

Rasmussen, J. and W. B. Rouse (eds.) (1980), "Human Detection and Diagnosis of System Failures", *NATO Symposium*, Roskilde, Denmark, Plenum Press.

Rich, E. and K. Knight (eds.) (1991), *Artificial Intelligence*, 2nd Edition, McGraw-Hill Inc..

Rich, S. H. and V. Venkatasubramanian (1989), "Causality-Based Failure-Driven Learning in Diagnostic Expert Systems", *AIChE Journal*, Vol. 35, No. 6, pp943-950.

Rumelhart, D. E. and J. L. McClelland (eds.) (1986), *Parallel Distributed Processing - Explorations in the Microstructure of Cognition*, Vol. 1: Foundations, The MIT Press, Cambridge, Massachusetts, London.

Rumelhart, D. E., G. E. Hinton and R. J. Williams (1986a), "Learning representation by back-propagation errors", *Nature (London)*, Vol. 323, pp533-536.

Saraiva, P. M. and G. Stephanopoulos (1992), "Continuous Process Improvement Through Inductive and Analogical Learning", *AIChE Journal*, Vol. 38, No. 2, pp161-183.

Scarl, E. A., J. R. Jamieson and C. I. Delaune (1987), "Diagnosis and Sensor Validation through Knowledge of Structure and Function", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 17, No. 3, pp360-368.

Shen, Q. and R. Leitch (1993), "Fuzzy Qualitative Simulation", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 23, No. 4, pp1038-1061.

- Shiozaki, J., H. Matsuyama, E. O'Shima and M. Iri (1985), "An Improved Algorithm for Diagnosis of System Failures in the Chemical Process", *Computer & Chemical Engineering*, Vol. 9, pp285-293.
- Simon, H. A. (1983), "Why should machines learn ?", in R. S. Michalski, J. G. Carbonell and T. M. Mitchell (eds.), *Machine Learning: An Artificial Intelligence Approach*, Palo Alto, CA: Tioga, pp25-37.
- Sorsa, T., H. N. Koivo and H. Koivisto (1991), "Neural networks in process fault diagnosis", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 21, No. 4, pp815-825.
- Sorsa, T. and H. N. Koivo (1993), "Application of artificial neural networks in process fault-diagnosis", *Automatica*, Vol. 29, No. 4, pp843-849.
- Steels, L. (1989), "Diagnosis with a Function-Fault Model", *Applied Artificial Intelligence*, Vol. 3, No. 2-3, pp213-237.
- Stengel, R. F. (1993), "Toward Intelligent Flight Control", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 23, No. 6, pp1699-1717.
- Stephanopoulos, G. (1990), "AI in Process Engineering Current and Future Trends", *Computers and Chemical Engineering*, Vol. 14, No. 11, pp1259-1270.
- Sugeno, M. (ed.) (1985), *Industrial applications of fuzzy control*, New York: North-Holland.
- Sussman, G. J. (ed.) (1975), *A Computer Model of Skill Acquisition*, American Elsevier, New York.
- Swartout, W. R. (1983), "XPLAIN: a system for creating and explaining expert consulting programs", *Artificial Intelligence*, Vol. 21, pp285-325.
- Takagi, H. (1995), "Applications of Neural Networks and Fuzzy Logic to Consumer Products", in J. Yen, R. Langari and L. A. Zadeh (eds.), *Industrial Applications of Fuzzy Logic and Intelligent Systems*, IEEE Press, pp93-105.



- Tsukamoto, Y. and T. Terano (1977), "Failure diagnosis by using fuzzy logic", *Proceedings of the IEEE Conference on Decision Making and Control*, New Orleans, pp1390-1395.
- Turban, E. (ed.) (1992), *Expert Systems and Applied Artificial Intelligence*, Macmillan Publishing Company, Inc.
- Ulieru, M. and R. Isermann (1993), "Design of a fuzzy-logic based diagnostic model for technical processes", *Fuzzy Sets and Systems*, Vol. 58, pp249-271.
- Vadera, S. (ed.) (1989), *Expert Systems Applications*, Sigma Press, 1989.
- Venkatasubramanian, V. and K. Chan (1989), "A Neural Network Methodology for Process Fault Diagnosis", *AIChE Journal*, Vol. 35, No. 12, pp1993-2002.
- Venkatasubramanian, V. and S. H. Rich (1988), "An Object-Oriented Two-Tier Architecture for Integrating Compiled and Deep-Level Knowledge for Process Diagnosis", *Computers and Chemical Engineering*, Vol. 12, No. 9/10, pp903-921.
- Wang, P. Z., H. M. Zhang and W. Xu (1990), "Pad-analysis of fuzzy control stability", *Fuzzy Sets and Systems*, Vol. 38, pp27-42.
- Wang, Z., M. T. Tham and A. J. Morris (1992), "Multilayer neural networks: approximated canonical decomposition of nonlinearity", *International Journal Control*, Vol. 56, pp655-672.
- Watanabe, K., I. Matsuura, M. Abe, M. Kubota and D. M. Himmelblau (1989), "Incipient fault diagnosis of chemical processes via artificial neural networks", *AIChE Journal*, Vol. 35, No. 11, pp1803-1812.
- Watanabe, K., S. Hirota, L. Hou and D. M. Himmelblau (1994), "Diagnosis of Multiple Simultaneous Fault via Hierarchical Artificial Neural Networks", *AIChE Journal*, Vol. 40, No. 5, pp839-848.
- Waterman, D. A. (ed.) (1985), *A Guide to Expert Systems*, Addison-Wesley.

- White, D. A. and D. A. Sofge (eds.) (1993), *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches*, Van Nostrand and Reinhold.
- Willis, M., C. D. Massimo, G. A. Montague, M. T. Tham and A. J. Morris (1991), "Artificial neural networks in process engineering", *IEE Proc.-D: Control, Theory and Applications*, Vol. 138, No. 3, pp256-266.
- Willsky, A. S. (1976), "A Survey of Design Methods for Failure Detection in Dynamic Systems", *Automatica*, Vol. 12, pp601-611.
- Wong, C. C., C. H. Chou and D. L. Mon (1993), "Studies on the output of fuzzy controllers with multiple inputs", *Fuzzy Sets and Systems*, Vol. 57, pp149-158.
- Yagawa, G., Y. Mochizuki and S. Yoshimura (1991), "Automated Structural Design Based on Expert's Knowledge and Fuzzy Control", *Computers in Engineering*, Vol. 1, The American Society of Mechanical Engineers, pp23-28.
- Yager, R. R. and L. A. Zadeh (eds.) (1994), *Fuzzy Sets, Neural Networks, and Soft Computing*, Van Nostrand Reinhold, New York.
- Yen, J., R. Langari and L. A. Zadeh (eds.) (1995), *Industrial Applications of Fuzzy Logic and Intelligent Systems*, IEEE Press.
- Zadeh, L. A. (1975), "The concept of a linguistic variable and its application to approximate reasoning", Parts 1, 2 and 3, *Inf. Science*, Vol. 8, pp199-249, 301-357, Vol. 9, pp43-80.
- Zadeh, L. A. (1979), "Fuzzy Sets and Information Granularity", in M. M. Gupta, R. K. Ragade and R. R. Yager (eds.), *Advances in Fuzzy Set Theory and Applications*, North-Holland Publishing Company, pp3-18.
- Zadeh, L. A., K. Fu, K. Tanaka and M. Shimura (eds.) (1975), *Fuzzy Sets and their Applications to Cognitive and Decision Processes*, Academic Press Inc..



- Zbikowski, R. and P. J. Gawthrop (1992), "A survey of neural networks for control", in K. Warwick, G. W. Irwin and K. J. Hunt (eds.), *Neural Networks for control and systems*, Peter Peregrinus Ltd on behalf of the IEE, pp31-50.
- Zhang, J. (1991), *Expert Systems in On-Line Process Control and Fault Diagnosis*, PhD Thesis, Control Engineering Centre, The City University.
- Zhang, J. and A. J. Morris (1994), "Fuzzy Neural Networks in Process Modelling and Fault Diagnosis", *Proceedings of ASI'94*, 27-1 July, Patras, Greece, Vol. 1, pp166-173.
- Zhang, J. and P. D. Roberts (1991a), "Process fault diagnosis with diagnostic rules based on structural decomposition", *J. Process Control*, Vol. 1, pp259-269.
- Zhang, J. and P. D. Roberts (1991b), "A Fault Diagnosis System with Self Reasoning Facility Based on Qualitative Modelling", in R. Hanus, P. Kool and S. Tzafestas (eds.), *Mathematical and Intelligent Models in System Simulation*, J. C. Baltzer AG, Scientific Publishing Co., IMACS, pp359-366.
- Zhang, J. and P. D. Roberts (1992a), "On-line process diagnosis using neural network techniques", *Trans. Inst. M. C.*, Vol. 14, No. 4, pp179-188.
- Zhang, J. and P. D. Roberts (1992b), "Use of genetic algorithms in training diagnostic rules for process fault diagnosis", *Knowledge-Based Systems*, Vol. 5, No. 4, pp277-288.
- Zhang, J., P. D. Roberts and J. E. Ellis (1988), "An Application of Expert Systems Techniques to the On-Line Control and Fault Diagnosis of a Mixing Process", *Journal of Intelligent and Robotic Systems*, Vol. 1, No. 3, pp209-223.
- Zimmermann, H. J. (1991), *Fuzzy set theory and its applications*, 2nd Edition, Kluwer Academic Press.